

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Институт Кибернетики
 Направление подготовки 15.03.06 Мехатроника и робототехника
 Кафедра систем управления и мехатроники

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Создание программной компоненты по распознаванию дорожных знаков в сложных погодных условиях на основе каскадного классификатора

УДК 656.055.9:004.932.72:551.515

Студент

Группа	ФИО	Подпись	Дата
8Е31	Кремлев Иван Андреевич		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. СУМ	Пякилля Б.И.	-		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. МЕН	Тухватулина Л.Р.	к.ф.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. ЭБЖ	Невский Е.С.	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Зав. кафедрой СУМ	Губин В.Е.	к.т.н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Код результата	Результат обучения (выпускник должен быть готов)
<i>Профессиональные компетенции</i>	
P1	Разрабатывать математические модели составных частей объектов профессиональной деятельности методами теории автоматического управления.
P2	Применять необходимые для построения моделей знания принципов действия и математического описания составных частей мехатронных и робототехнических систем (информационных, электромеханических, электрогидравлических, электронных элементов и средств вычислительной техники).
P3	Выполнять расчетно-графические работы по проектированию информационных, электромеханических, электрогидравлических, электронных и микропроцессорных модулей мехатронных и робототехнических систем.
P4	Разрабатывать конструкторскую проектную документацию электрических и электронных узлов (включая микропроцессорные) мехатронных и робототехнических систем, принципиальные электрические схемы, печатные платы, схемы размещения, схемы соединения.
P5	Участвовать в проведении предварительных испытаний составных частей опытного образца мехатронной или робототехнической системы по заданным программам и методикам и вести соответствующие журналы испытаний
<i>Универсальные компетенции</i>	
P6	Обладать способностью владеть культурой мышления, способен к обобщению, анализу, восприятию информации, постановке цели и выбору путей ее достижения
P7	Обладать способностью уметь использовать нормативные правовые документы в своей деятельности.
P8	Обладать способностью иметь навыки работы с компьютером как средством управления информацией.
P9	Обладать способностью владеть одним из иностранных языков на уровне не ниже разговорного.

Министерство образования и науки Российской Федерации
 Федеральное государственное бюджетное образовательное учреждение
 высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**



Институт Кибернетики
 Направление подготовки (специальность) 15.03.06 Мехатроника и робототехника
 Кафедра систем управления и мехатроники

УТВЕРЖДАЮ:

Зав. кафедрой

 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8Е31	Кремлеву Ивану Андреевичу

Тема работы:

Создание программной компоненты по распознаванию дорожных знаков в сложных погодных условиях на основе каскадного классификатора	
Утверждена приказом директора (дата, номер)	№3935/с от 05.06.2017

Срок сдачи студентом выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе</p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<ol style="list-style-type: none"> 1. Существующие алгоритмы распознавания дорожных знаков. 2. Требование реализовать программную компоненту по распознаванию дорожных знаков.
<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсужде-</i></p>	<ol style="list-style-type: none"> 1. Провести обзор имеющихся решений. 2. Выбрать технологии и подходы для собственного алгоритма распознавания дорожных знаков. 3. Реализовать алгоритм с помощью библиотеки компьютерного зрения OpenCV.

ние результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).	4. Протестировать программную компоненту в реальных условиях.
Перечень графического материала	Презентация на 11 слайдах
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Тухватулина Лилия Равильевна
Социальная ответственность	Невский Егор Сергеевич

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. СУМ	Пякилля Б.И.	-		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8Е31	Кремлев Иван Андреевич		

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**



Институт Кибернетики

Направление подготовки (специальность) 15.03.06 Мехатроника и робототехника

Уровень образования - бакалавриат

Кафедра систем управления и мехатроники

Период выполнения весенний семестр 2016/2017 учебного года

Форма представления работы:

Бакалаврская работа
(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
20.05.2017	Основная часть	75
25.05.2017	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
01.06.2017	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. СУМ	Пякилля Б.И.	-		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Зав. кафедрой СУМ	Губин В.Е.	к.т.н.		

РЕФЕРАТ

Выпускная квалификационная работа состоит из 77 страниц, включает в себя 16 рисунков и 16 таблиц. При работе были использованы 30 источников литературы, из которых 2 ссылки на Web-страницы Internet.

Ключевые слова: ДОРОЖНЫЕ ЗНАКИ, OPENCV, ВЫДЕЛЕНИЕ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ, МАШИННОЕ ОБУЧЕНИЕ.

Цель работы: разработать программную компоненту для автоматического распознавания дорожных знаков в сложных погодных условиях.

Результатом работы является приложение, способное детектировать дорожные знаки на последовательности видеок кадров.

Объектом разработки является система автоматического распознавания дорожных знаков.

Приложению для корректной работы требуется библиотека OpenCV.

Областью применения данной работы является процесс паспортизации автомобильных дорог. Прогнозом дальнейшего развития является улучшение и расширение реализованного приложения.

СОДЕРЖАНИЕ

Введение	11
1. Обзор имеющихся решений	13
1.1 Методы, основанные на поиске формы знака	14
1.2 Методы, основанные на цвете знака	15
1.3 Методы, основанные на машинном обучении	16
1.3.1 Нейронные сети.....	17
1.3.2 Метод опорных векторов	17
1.3.3 Деревья принятия решений.....	18
1.3.4 Каскадный классификатор	18
2. Технологии и подходы.....	19
2.1 Примитивы Хаары.....	20
2.2 Интегральное представление изображений.....	22
2.3 AdaptiveBoosting	23
2.4 Фильтр Гаусса.....	25
2.5 Детектор границ Кенни.....	27
2.6 Преобразование цветового пространства	28
3. Практическая реализация.....	29
3.1 Инструментарий разработки	29
3.2 Реализация алгоритма компоненты.....	32
3.3 Выделение элементов знака	34
3.4 Создание классификатора	35
3.5 Достигнутые результаты	42
4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	48
4.1 Оценка коммерческого потенциала, перспективности и альтернатив проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	48
4.1.1 Методы коммерциализации результатов инженерных решений.....	48
4.1.2 SWOT-анализ	49
4.2 Планирование и формирование бюджета научных исследований	50

4.2.1 Планирование научно-исследовательских работ	50
4.2.2 Определение трудоемкости выполнения работ	51
4.2.3 Разработка графика проведения научного исследования	52
4.2.4 Расчет материальных затрат НТИ	56
4.2.5 Расчет затрат на основные средства	58
4.2.6 Основная заработная плата исполнителей темы	58
4.2.7 Дополнительная заработная плата исполнителей темы	61
4.2.8 Отчисления во внебюджетные фонды	61
4.2.9 Накладные расходы	62
4.2.10 Формирование бюджета затрат научно-исследовательского проекта	63
4.2.11 Проведение анализа безубыточности проекта	64
4.3 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	65
4.3.1 Определение ресурсной, финансовой и экономической эффективности исследования	65
5. Социальная ответственность	69
5.1 Экологическая безопасность	70
5.2 Электромагнитное излучение	71
5.3 Работа в условиях чрезвычайных ситуаций	72
5.4 Искусственный интеллект	73
5.5 Правовые и организационные вопросы обеспечения безопасности	75
Заключение	76
Список использованной литературы	77

Определения

В бакалаврской работе использовались следующие термины с соответствующими определениями:

Машинное обучение – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решению множества сходных задач.

RGB – аддитивная цветовая модель, как правило, описывающая способ кодирования цвета для цветопроизводства.

HSV – цветовая модель, в которой координатами цвета являются цветовой тон, насыщенность и значение цвета.

Обозначения и сокращения

ИИ – искусственный интеллект.

CUDA – это архитектура параллельных вычислений от NVIDIA, позволяющая существенно увеличить вычислительную производительность благодаря использованию графических процессоров.

Введение

За последние несколько десятилетий транспортные средства настолько сильно интегрировались в нашу повседневную жизнь, что сейчас трудно представить жизнедеятельность людей без них. Поэтому, общество старается постоянно улучшать данную отрасль не только с практической, но и с научной стороны. С 1980 года количество личных автомобилей в мире увеличилось на 84%, что является следствием быстрого роста населения и стремительного темпа жизни. С ростом числа автомобилей возросло и количество дорожно-транспортных происшествий. [1] Управление транспортным средством требует большого внимания и осторожности от водителя. Ежедневно на дорогах из-за невнимательности водителей происходят аварии, которые нередко заканчиваются травмами и даже гибелью людей. Поэтому, если бы имелось такое транспортное средство, которое позволяло бы человеку спокойно и безопасно добираться до необходимого места, одна из важных проблем для человечества была бы решена. При современном развитии технологий, данные мечты уже сейчас становятся реальностью. Желание добиться улучшения ситуации дорожного движения за счет автоматизации подтолкнуло ученых к разработке автономных автомобилей, способных передвигаться без участия человека.

На текущий момент уже существуют прототипы практически всех видов беспилотных транспортных средств. Наиболее активно используются военные и промышленные роботизированные транспортные устройства. Но наука не стоит на месте и уже сейчас актуальными являются разработки беспилотных легковых и грузовых автомобилей, роботизированной авиации, водных видов транспортных средств, а также мотоциклы и даже велосипеды.

Также одним из перспективных направлений в этой области является содержание и обслуживание дорог без участия человека. Каждая автомобильная трасса должна периодически проверяться на наличие поврежденных знаков, так как они представляют угрозу безопасности участников дорожного движения. Такие проверки делаются человеком, который вручную записывает проблемные объекты. Задача ручной проверки является утомительной и кроме того, имеет

место человеческая ошибка. При помощи методов компьютерного зрения, данная задача может быть автоматизирована и выполняться более часто, что приведет к повышению безопасности на дорогах. [2]

Но одной из наиболее трудных задач, которые стоят перед исследователями, является распознавание всевозможных объектов, которые встречаются на пути у транспортного средства в реальном времени.

За последние 20 лет многие ученые из разных стран пытались решить задачу распознавания дорожных знаков. Самые первые исследования на данную тематику появились в 80-х годах XX века. Однако, вычислительные мощности на тот момент не позволяли удовлетворительно реализовать обработку видеоряда. Кроме того, не существовало камер, которые позволяли получать изображения необходимого формата и размера. Но на сегодняшний день развитие технического уровня современных мобильных процессоров достигло такого уровня, что производители автомобилей могут устанавливать системы по распознаванию дорожных знаков в серийные автомобили. Например, технология CUDA от NVIDIA, которая значительно увеличивает вычислительную производительность за счет использования архитектуры параллельных вычислений, позволяет сокращать время на обработку изображений до нескольких миллисекунд, что удовлетворяет требованиям по обработке видеоряда в режиме реального времени. [3] Но системы, которые используются в серийных автомобилях имеют низкую точность распознавания, а также не всегда корректно обнаруживают отечественные дорожные знаки.

Некоторые люди ошибочно полагают, что задача распознавания дорожных знаков является тривиальной, ведь все знаки стандартизированы. Действительно, в настоящее время разработчики довольно успешно решают задачу, связанную с распознаванием дорожных знаков по изображению. В 2011 году на базе 50 тысяч изображений немецких дорожных знаков, специалисты из университета Лугано создали программу по их распознаванию на основе нейронных сетей. На конкурсе она показала результат 99,46%, опередив не только другие программы, но даже лучшего из 32 людей-участников конкурса (99,22%), при том, что средний показатель среди людей составил 98,84%. [4]

Но в системах помощи водителю стоит задача распознавания дорожного знака не по изображению, а по видеоряду в реальном времени. В связи с этим возникает проблема – как достоверно определять объекты и при этом исключить ложные срабатывания. Для большей наглядности данной проблемы приведем следующий пример. Один час видео со скоростью 24 кадра в секунду состоит из 86400 кадров. Полагаем, что каждую минуту нам встречается знак, который охватывает около 40 кадров, тогда получим, что из всех кадров за час лишь чуть больше 2 тысяч содержат изображения дорожных знаков, а оставшиеся 84 тысячи не содержат ничего и не имеют практической пользы для решения данной задачи. Также стоит отметить недостаток данных систем для внедрения в коммерческое производство. Оборудование автомобилей дополнительными камерами и датчиками значительно повышает стоимость автомобиля для потребителя.

Целью данной работы является решение задачи распознавания дорожных знаков в сложных погодных условиях, а также проверка эффективности и удобства использования реализованных алгоритмов. Для получения исходных данных будет применяться камера, закрепленная в автомобиле — типовой видеорегистратор, обеспечивающий видеосъемку в формате Full-HD.

1. Обзор имеющихся решений

Выбрать наилучший метод распознавания дорожных знаков довольно затруднительно, т.к. имеется множество факторов, которые влияют на точность распознавания. Среди основных, которые больше всего влияют на точность распознавания относятся следующие:

1. Представление информации – статическое изображение или видеофрагмент.
2. Универсальность программы – использование для определенного класса дорожных знаков или для всех.
3. Условия съемки – солнечный день, ночью или в условиях дождя и тумана.
4. Качество оборудования – камера с высоким или низким разрешением.
5. Режим обработки информации – в реальном времени или в оффлайн-режиме?

Все эти факторы являются важными для выбора метода. Например, метод, основанный на определении цветного изображения бесполезен, если мы используем черно-белую камеру. Но с другой стороны, он подходит для распознавания, если мы используем видеоряд, снятый в солнечный день в высоком разрешении. Все методы по распознаванию дорожных знаков можно разделить на несколько групп, перейдем к их обзору.

1.1 Методы, основанные на поиске формы знака

Наиболее распространенный подход в обнаружении дорожных знаков основан на нахождении области изображения, который содержит интересующий нас цвет. На полученных областях определяются либо дорожные знаки, либо области, где они могут гипотетически располагаться. Для этого может использоваться пороговая обработка изображений или более продвинутые методы сегментации изображения. Сегментацией называют процесс деления целого изображения на несколько сегментов или множество пикселей, которые еще называют суперпикселями. Целью сегментации является упрощение представления изображения для последующего анализа. Обычно сегментацию используют для выделения объектов и границ (кривые, линии и т.п.) на изображениях. В ходе выполнения сегментации изображений происходит процесс присвоения таких меток каждому пикселю изображения, что пиксели с одинаковыми метками обладают общими визуальными характеристиками. Пороговая обработка является наиболее простым методом сегментации, что привлекает к нему большое внимание ученых. Данный метод ориентирован на обработку изображений, отдельные однородные участки которых различаются средней яркостью. Самым простым, но при этом наиболее часто применяемым видом сегментации называют бинарную сегментацию, когда имеется только два типа однородных участков. Исследователи много экспериментировали с сегментацией и пороговой обработкой. Полученные результаты говорят о хорошей точности распознавания дорожных знаков в подобных системах. [5]

Еще одним методом из данной группы является преобразование Хафа. Оно используется для поиска объектов, которые принадлежат строго определённой

классу фигур. Важным моментом в данном методе является использование процедуры голосования, которая применяется к пространству параметров, из которого и получаются объекты определённого класса фигур по локальному максимуму. При вычислении трансформации Хафа получается так называемое, накопительное пространство.

Преобразование Хафа базируется на представлении объекта в виде параметрического уравнения. Параметры этого уравнения представляют фазовое или аккумуляторное пространство. Затем, берётся двоичное изображение, например, результат работы детектора границ Кенни. Все точки границ перебираются и делается предположение о том, что точка принадлежит линии искомого объекта. Таким образом, мы рассчитываем нужное уравнение для каждой точки изображения и получаем необходимые параметры, которые сохраняются в пространстве Хафа. Заключительным шагом является обход пространства Хафа и выбор его максимальных значений, за которые «проголосовало» больше всего пикселей картинки, что и даёт нам параметры для уравнений искомого объекта. [6]

1.2 Методы, основанные на цвете знака

Главный недостаток такого подхода заключается в том, что в зависимости от времени суток, погодных условий, теней и т.д. цвет знака может быть искажен. Стандартное RGB цветовое пространство очень чувствительно к освещению, поэтому исследователи стараются использовать другие цветовые пространства, такие как HSI или LAB. Но тем не менее, на основе данного подхода было создано довольно много разнообразных методов.

Альберто Бругги предложил способ, основанный на зависимости от источника света. Он состоит в том, чтобы найти белый объект в сцене и вычислить различие между изображением, белым и теоретически белым цветом. В реальных условиях нельзя рассчитывать на наличие белого ориентира, т.к. дорога обычно серая. Поэтому, предполагаем, что дорога должна быть серой. Затем выполняется цветное выравнивание с линеаризацией гамма функции. [7]

Серхио Эскалера предложил подход для обнаружения красного в цветовом пространстве HSI. Входное изображение сначала преобразовывается из цвето-

вого пространства RGB в HSI. Для каждого пикселя значение оттенка и насыщенности вычисляется так, чтобы диапазон красного был ярко выражен. Авторы предполагают, что значение оттенка и насыщенность измерена в диапазоне от 0 до 255. Полученные значения перемножаются; авторы заявляют, что подобный подход позволяет обезопаситься в случае ошибки нахождения одной из величин. [8]

Ксиоли Хао в своих исследованиях использовал цветовую модель CIECAM97. Изображения сначала преобразуются из RGB к CIE, и затем к LCH. Авторы рассматривают четыре случая: среднестатистические условия, а также условия во время солнечного, облачного и дождливого дня. Используя приемлемые диапазоны, изображение рекурсивно разделяется на сектора, благодаря чему достигается predetermined размер пикселя. [9]

Результаты подобных исследований говорили о высоком потенциале данного подхода, но различные факторы, такие как изменение освещения, тени, неблагоприятные погодные условия значительно влияют на процент детектирования. Несмотря на это, использование методов по распознаванию дорожных знаков имеет место быть, как дополнительный фактор в алгоритме.

1.3 Методы, основанные на машинном обучении

В настоящее время набирают обороты разнообразные методы машинного обучения. Алгоритмы машинного обучения находят широкое применение в системах различного назначения: поисковых системах, алгоритмах распознавания, анализа и синтеза речи, медицинской диагностике, биоинформатике, финансовом прогнозировании и т.д. Исключением не является и компьютерное зрение. Например, подавляющее большинство современных систем детектирования объектов на изображениях и видео основано на применении алгоритмов машинного обучения. Такой подход позволяет компьютерной системе самой «научиться» отличать изображения, содержащие искомый объект, от остальных, используя для этого лишь примеры таких изображений. Также кластеризация и обучение с учителем успешно применяется в алгоритмах классификации изображений, опять же позволяя автоматически установить неявные различия между изображениями различных типов.

Наиболее популярными являются следующие алгоритмы машинного обучения.

1.3.1 Нейронные сети

Нейросетевые методы включают в себя целый класс алгоритмов. Основная идея, которая лежит в их основе заключается в последовательном преобразовании сигнала параллельно работающими функциональными элементами, или как их еще называют, нейронами. Процесс обучения таких нейронных сетей сводится к уменьшению среднеквадратичной ошибки. Системы обнаружения объектов на изображениях, основанные на нейронных сетях, используют иерархическую структуру. Сначала вектор признаков обрабатывается грубой сетью с высоким уровнем ошибок второго рода, а затем, если вектор не был классифицирован как не объект, решение корректируется более точной, но при этом и более медленной сетью. Достоинством нейронных сетей является высокая точность обнаружения при правильной настройке параметров сети. Из недостатков можно выделить сильную чувствительность к шуму, необходимость в тщательной настройке параметров нейронной сети для получения хороших результатов. Самым же главным недостатком является высокая вычислительная сложность, и, как следствие, скорость работы, которая недостаточна для обработки в реальном времени.

1.3.2 Метод опорных векторов

Это набор схожих алгоритмов вида «обучение с учителем», который используется при решении задач классификации и регрессионного анализа. Данный метод принадлежит к семейству линейных классификаторов. Суть метода заключается в нахождении гиперплоскости, которая разделяет два класса. Из всех возможных гиперплоскостей, разделяющих два класса, нужно выбрать такую гиперплоскость, расстояние до которой от каждого класса максимально. Эту гиперплоскость называют оптимальной разделяющей гиперплоскостью, а соответствующий ей линейный классификатор называется оптимально разделяющим классификатором. К достоинствам метода опорных векторов относят высокую устойчивость к переобучению, возможность уменьшения чувствительности к шуму за счет снижения точности. При этом, поиск оптимального соотношения

данных параметров требует точной настройки. А также точность работы метода уступает другим методам машинного обучения.

1.3.3 Деревья принятия решений

Это один из алгоритмов принятия решений, который используется для различных моделей. Структура дерева состоит из ребер и узлов: на ребрах («ветках») дерева решения записаны атрибуты, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах — атрибуты, по которым различаются объекты. Чтобы классифицировать новый объект, необходимо спуститься по дереву до листа и выдать соответствующее значение. Подобные деревья решений широко используются в интеллектуальном анализе данных. Достоинствами алгоритма являются: высокая скорость обучения, высокое качество получаемых моделей (сравнимое с нейронными сетями и ансамблями нейронных сетей). Вместе с тем, можно отметить и недостатки алгоритма: необходимость в большом количестве памяти. Также обученная модель работает несколько медленнее других алгоритмов (если в модель входит 100 деревьев, мы должны пройти по всем, чтобы получить результат), что делает данный алгоритм непригодным для поиска объектов в видеоряде в реальном времени.

1.3.4 Каскадный классификатор

Каскадный классификатор или метод Виола-Джонса был предложен Полом Виолой и Майклом Джонсом и стал первым методом, который демонстрировал высокую точность распознавания объектов на изображениях в реальном времени. В данном алгоритме используется набор признаков, близких к признакам Хаара совместно с одной из вариаций алгоритма AdaBoost. AdaBoost (Adaptive Boosting) – алгоритм усиления классификаторов путем объединения их в отдельные группы. Впервые данный алгоритм был предложен Йоавом Фройндом и Робертом Шапире. Обычно он используется в сочетании с другими алгоритмами классификации для улучшения их эффективности. Данный алгоритм называется адаптивным потому, что каждая следующая группа классификаторов строится по объектам, неверно классифицированным предыдущими группами. Достоинствами подобного метода являются высокая скорость обнаружения за счет использования каскадного классификатора, высокая точность обнаружения

с низким процентом ложных срабатываний, в целом сравнимая с точностью обнаружения у гораздо более медленных алгоритмов. Кроме того, алгоритм подстраивается под проблемные элементы из обучающей выборки и теоретически метод достигает нулевой ошибки обучения за конечное число итераций. Но при этом данному алгоритму требуется большая обучающая выборка из позитивных и негативных изображений и большое время обучения. Если же мы имеем в наличии достаточные вычислительные мощности, то данный фактор не является решающим. [10]

Таким образом, наилучшим вариантом будет являться алгоритм, основанный на методе машинного обучения с использованием дополнительных факторов для улучшения точности распознавания дорожных знаков в виде перехода в другое цветовое пространство, а также поиска границ контура знака.

2. Технологии и подходы

Проведя обзор имеющихся решений, можно приступить к созданию собственного алгоритма. Обнаружение дорожных знаков будет выполняться в два этапа: локализация знака и предобработка изображения, а уже затем последующая классификация. В ходе решения задачи распознавания дорожных знаков важно найти наиболее оптимальное сочетание сложности вычислений и желаемой точности. Это связано с тем, что задачей нашего проекта является распознавание дорожных знаков в режиме реального времени. В данном разделе бакалаврской работы будут описаны все этапы обработки кадров видеопотока, а также использующиеся в них методы и алгоритмы.

Как отмечалось ранее, в своей работе мы будем использовать метод Виолы-Джонса, который является одним из основных методов поиска объектов на изображении в реальном времени. Выделим основные принципы, на которых основана работа данного метода:

1. Поиск дорожных знаков с помощью признаков Хаара.
2. Интегральное представление изображений.
3. Каскадная классификация с применением бустинга.
4. Применение фильтра Гаусса
5. Применение детектор границ Кенни.

6. Преобразование цветового пространства.

Разделяющие свойства признаков являются самыми важными критериями качества признаков при решении широкого спектра задач, связанных с распознаванием. Также следует учитывать необходимость того, что область нужного нам объекта важно найти как можно быстрее. Данного результата можно добиться только с учетом классификации большого числа элементов. Для решения задач, которые связаны с распознаванием наиболее рационально использовать относительно простые алгоритмы получения признаков, например, использование алгоритмов распознавания дорожных знаков на основе примитивов Хаара. Перейдем к рассмотрению принципов работы примитивов Хаара.

2.1 Примитивы Хаара

Признаком f объекта a называют отображение $f:A \rightarrow Df$, где Df - множество допустимых значений признака. Если задан набор векторов f_1, \dots, f_n , тогда вектор $x = (f_1(a), \dots, f_n(a))$ называется признаковым описанием объекта $a \in A$. [11] Впервые использование для обнаружения объектов признаков, основанных на вейвлетах Хаара, было предложено в работе Папагеоргиу. [12] Виола и Джонс адаптировали эту идею в своей работе и получили прямоугольные признаки, названные признаками Хаара. [13] Внешний вид данных признаков можно увидеть на рис.1.

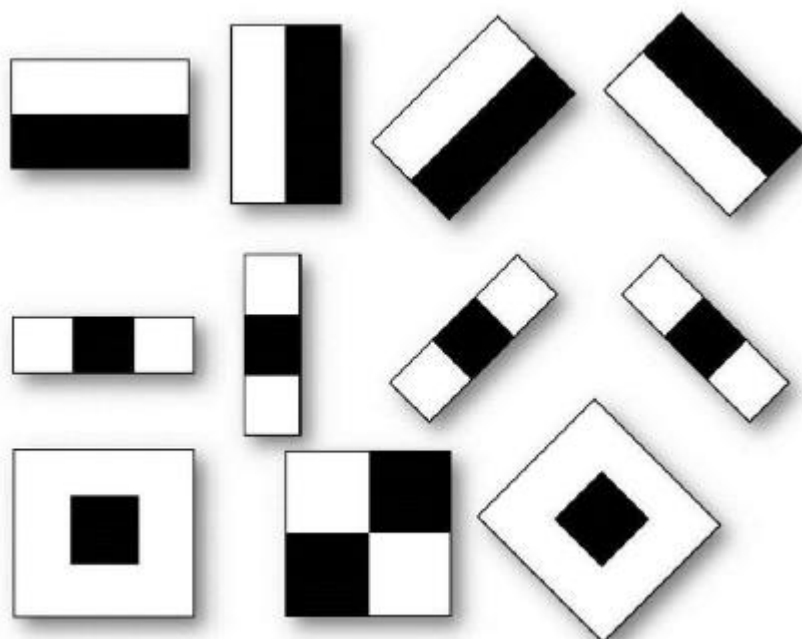


Рис.1 Признаки Хаара

В расширенном методе Виолы-Джонса, представленном в библиотеке компьютерного зрения OpenCV, и используемом в разрабатываемой системе, используются также дополнительные признаки, представленные на рис.2.

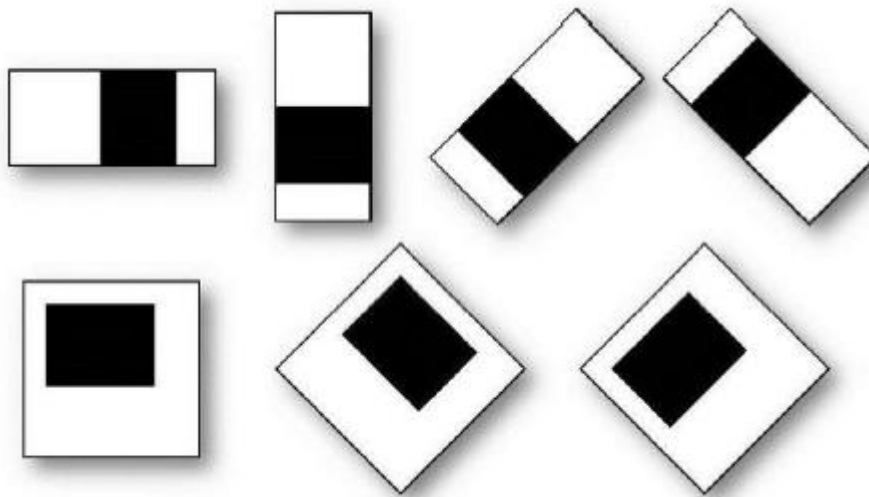


Рис. 2 Дополнительные признаки Хаара

Результатом вычисления такого признака на интегральном представлении изображения будет:

$$F = U - V$$

где U – сумма значений яркостей точек, закрываемых светлой частью признака, а V – сумма значений яркостей точек, закрываемых темной частью признака.

Такие признаки описывают описание перепада яркости по обеим осям изображения. Поиск дорожных знаков происходит при помощи так называемого сканирующего окна. Окно перемещается по изображению с шагом в 1 пиксель и для каждого его положения вычисляются признаки Хаара с разным масштабом и положением в окне. При этом само сканирование проводится также и для разных масштабов сканирующего окна. Найденные признаки передаются классификатору, который определяет по их значениям, является ли область изображения, соответствующая окну, знаком или нет. [14]

2.2 Интегральное представление изображений

Для расчета яркости прямоугольного участка изображения используется так называемое интегральное представление. Данное представление часто применяется и во многих других разработанных алгоритмах компьютерного зрения. Интегральное представление позволяет быстро рассчитывать суммарную яркость произвольного прямоугольника на заданном изображении, причем время расчета не зависит от площади прямоугольника. Интегральным оно называется по аналогии с математической операцией интегрирования, вычисляющей сумму значений функции на отрезке. В данном случае функцией является яркость пикселя $I(x,y)$.

Интегральное представление изображения представляет собой матрицу, размеры которой совпадают с размерами исходного изображения. В каждом элементе такой матрицы хранится сумма интенсивностей всех пикселей, находящихся левее и выше данного элемента. Элементы матрицы рассчитываются в соответствии со следующей формулой:

$$I(x, y) = \sum_{x \leq x, y \leq y} i(x, y)$$

где $I(x,y)$ – значение точки (x,y) интегрального изображения; $i(x,y)$ – значение интенсивности исходного изображения. [15]

Расчет интегральной матрицы производится за один проход и занимает линейное время относительно числа пикселей в изображении. Расчет возможен по следующей формуле:

$$I(x, y) = Im(x, y) - I(x - 1, y - 1) + I(x, y - 1) + I(x - 1, y)$$

Построив интегральную матрицу, можно очень быстро вычислять сумму пикселей произвольного прямоугольника произвольной площади. Пусть в прямоугольнике ABCD есть интересующий нас объект D.

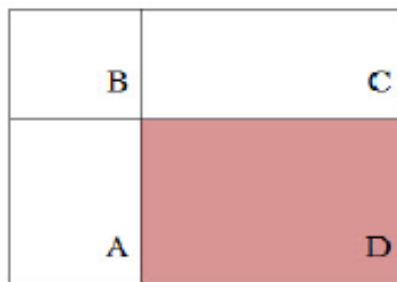


Рис.3 Прямоугольник с областями

Из рисунка очевидно, что сумму пикселей в прямоугольнике D можно выразить посредством суммы и разности смежных прямоугольников по следующей формуле:

$$S(D) = I(B) + I(ABCD) - I(BA) - I(BC)$$

Таким образом, значение любого признака Хаара можно вычислить посредством всего нескольких операций, количество которых зависит только от количества прямоугольников в признаке и не зависит от площади исследуемой области.

2.3 AdaptiveBoosting

Одним из наиболее распространенных подходов к решению задачи распознавания дорожных знаков является усиление простых классификаторов. Данный подход базируется на комбинировании нескольких простых классификаторов в один, но при этом более сильный. Эффективность классификатора заключается в параметре, который называется сила, что означает качество решения поставленной задачи классификации. В данном разделе дипломной работы описано семейство алгоритмов, на основе которых был создан алгоритм AdaBoost (от английских слов «adaptive — адаптивность» и «boosting — усиление»). Данный алгоритм был успешно апробирован для решения огромного круга задач, связанных с распознаванием на изображении, в том числе и для распознавания дорожных знаков. Этот подход в настоящее время активно применяется во многих областях и продолжают как теоретические, так и прикладные исследования алгоритмов AdaBoost. В числе основных преимуществ методов AdaBoost можно отметить:

- Высокая скорость работы;
- Высокая эффективность (точность);
- Простота реализации.

Методы распознавания, основанные на усилении простых классификаторов, базируются на следующей идее: совместить несколько элементарных признаков для того, чтобы скомбинированный признак получился более сильным. [16]

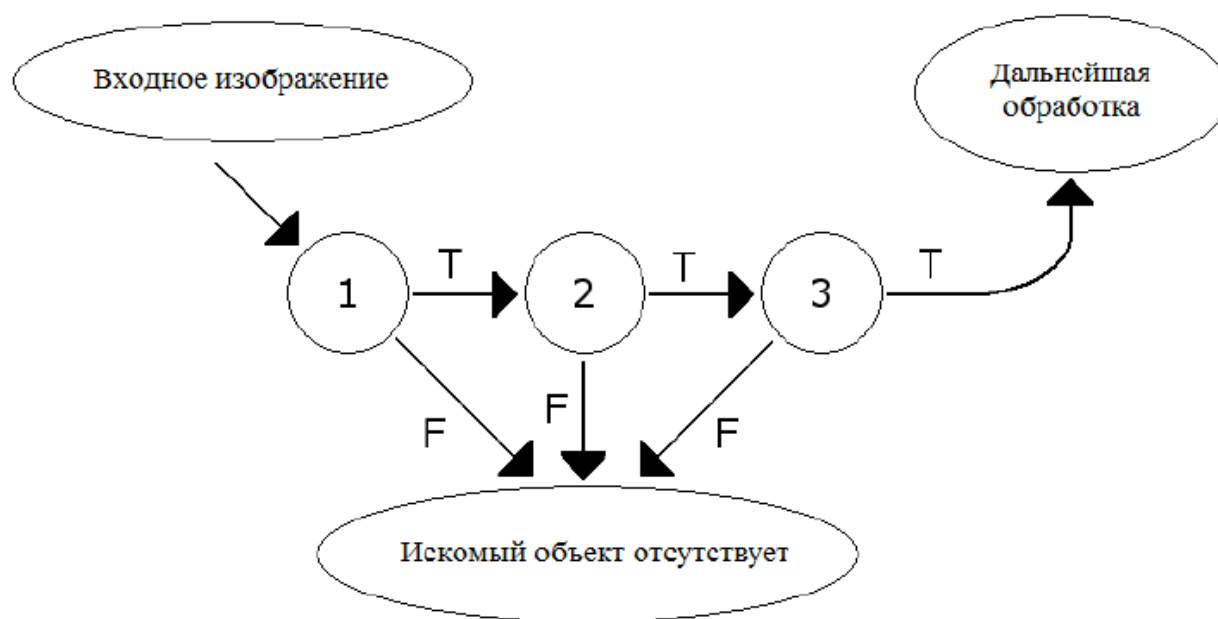


Рис. 4 Каскадный классификатор

Для большей понятности вышесказанного, рассмотрим пример с автомобильными гонками. Допустим, человек, который интересуется автомобильным спортом, решил разработать программу, которая помогала бы определять победителя и предсказывать шансы на победу различных гонщиков. В ходе просмотра определенного числа соревнований и проведя опрос среди зрителей, делающих ставки, этот человек может выделить несколько эмпирических правил:

- Нужно ставить на машину, которая победила в трех предыдущих кругах;
- Нужно ставить на ту машину, которую поддерживают максимальным количеством ставок;

– Фаворитами всегда являются победители предыдущих гонок и т. д.

Логично предположить, что использование каждого из вышеперечисленных правил по отдельности нецелесообразно. Поэтому, возникает необходимость наиболее рационально скомпоновать данные правила для достижения наилучшего результата. Таким образом, набор алгоритмов, работающих с использованием метода усиления простых классификаторов, позволяет собрать более сильный признак из нескольких простых. В результате классификации будет получен набор областей изображения, содержащих искомый объект. Затем исключаются вложенные повторения при обнаружении одного и того же объекта, вызванные масштабированием сканирующего окна. Для дальнейшей обработки найденные знаки переводятся в градации серого и масштабируются до необходимого размера. [17]

2.4 Фильтр Гаусса

С целью устранения шумов на изображениях знаков будем использовать фильтр Гаусса. Фильтр Гаусса – это фильтр размытия изображения, который использует нормальное распределение (также называемое Гауссовым распределением) для вычисления преобразования, применяемого к каждому пикселю изображения. Нормальное распределение для двух измерений описывается формулой (3.3) [18]

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

где r – радиус размытия, σ – стандартное отклонение распределения Гаусса.

Данная формула задает поверхность, имеющую вид концентрических окружностей с нормальным распределением от центральной точки. Пиксели, где распределение отлично от нуля используются для построения матрицы свертки, которая применяется к исходному изображению. Значение каждого пикселя становится средне взвешенным для окрестности. Исходное значение пикселя принимает наибольший вес (имеет наивысшее Гауссово значение), и соседние пиксели принимают меньшие веса, в зависимости от расстояния до них. Пример работы размытия по Гауссу для одномерного массива можно увидеть на рис.5.

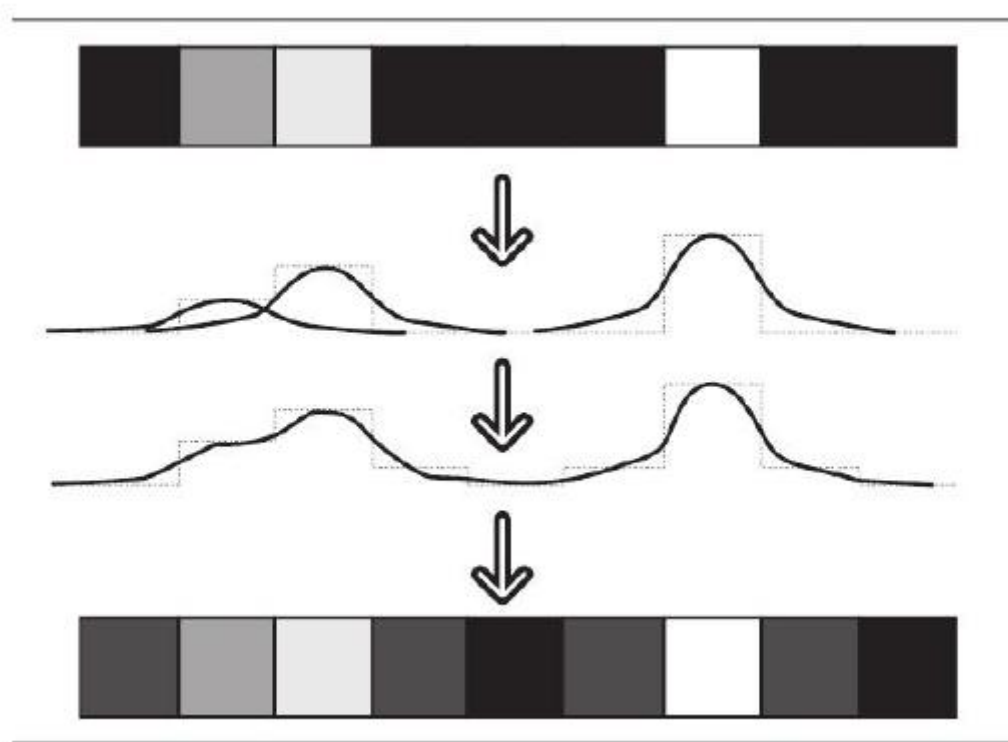


Рис. 5 Размытие по Гауссу на одномерном массиве

Таким образом, размытие по Гауссу позволит нам избавиться от нежелательных шумов на изображениях, что сведет к минимуму их влияние при дальнейшей классификации знаков. Результат применения фильтра Гаусса к целому изображению продемонстрирован на рис.6.

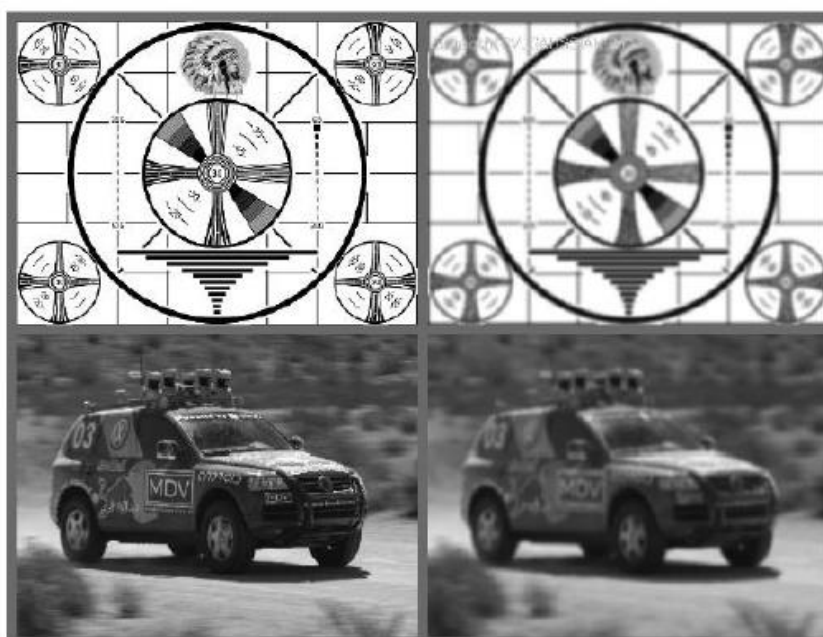


Рис. 6 Результат применения фильтра Гаусса

2.5 Детектор границ Кенни

В 1953 г. Джон Кенни представил решение математической проблемы получения фильтра, который был бы оптимален по критериям локализации, выделения и минимизации нескольких откликов одного края. Это значит то, что детектор будет реагировать на границы, но при этом игнорировать ложные, точно определять линию границы и реагировать на каждую границу только один раз. Это позволит избежать восприятия широких полос изменения яркости как совокупности границ. Кенни ввел термин «подавление немаксимумов», которое означает, что пикселями границ объявляются точки, в которых достигается локальный максимум градиента в направлении вектора градиента. [19]

Несмотря на то, что его работы были проведены еще в 1986 году, на заре компьютерного зрения, детектор границ Кенни до сих пор является одним из наилучших детекторов. Принцип работы детектора базируется на пяти основных частях.

1. *Сглаживание* - размытие изображения, которое предназначено для удаления шумов. Оператор Кенни использует фильтр, который может быть приближен к первой производной гауссианы.

2. *Поиск градиентов*. В данном пункте, границы отмечаются в тех местах, где градиент изображения приобретает наибольшее значение. Они могут иметь различное направление, поэтому алгоритм Кенни использует четыре фильтра для обнаружения горизонтальных, вертикальных и диагональных ребер в размытом изображении. Угол направления вектора градиента округляется и может принимать такие значения: 0, 45, 90, 135.

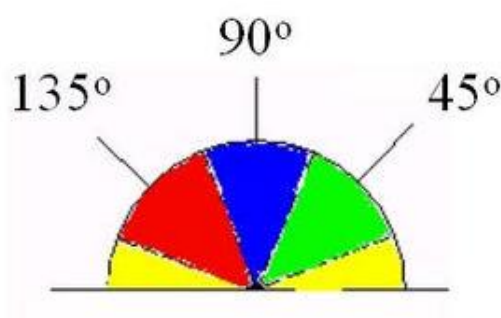


Рис. 7 Направление градиента

3. *Подавление немаксимумов.* В данном пункте как границы отмечаются только локальные максимумы.
4. *Двойная пороговая фильтрация.* Потенциальные границы определяются пороговыми.
5. *Трассировка области неоднозначности.* Итоговые границы определяются путём подавления всех краёв, не связанных с сильными границами.

Перед применением детектора обычно преобразуют изображение в оттенки серого, чтобы уменьшить вычислительные затраты. Этот этап характерен для многих методов обработки изображений

2.6 Преобразование цветового пространства

Специфические условия освещённости значительно влияют на возможность корректного восприятия цвета объектов снимаемой сцены. При съёмке реальной дорожной ситуации на знаках возникает целый ряд различных условий освещения, сильно усложняющих детектирование объектов: прямые солнечные лучи, отражённый свет, затенения, свет автомобильных фар в тёмное время суток. Более того, различные искажающие эффекты могут присутствовать на одном дорожном знаке одновременно. Таким образом, на изображениях, получаемых с камеры, установленной в автомобиле, не всегда возможно выделить интересующую область по цвету с помощью простого применения порогового фильтра напрямую в цветовом пространстве RGB (Red, Green and Blue). Для извлечения красного цвета из входного изображения необходимо использовать информацию о цвете каждого пикселя независимо от его неконтролируемого показателя освещённости. Для этих целей было выбрано цветовое пространство HSV (Hue, Saturation and Value). Цветовое пространство HSV принадлежит к группе цветовых пространств, изначально основанных на человеческом восприятии цветов. HSV в основном используется в области компьютерного зрения и компьютерной графики. Рис. 8 иллюстрирует стандартное представление цветового пространства HSV в виде конуса. [20]

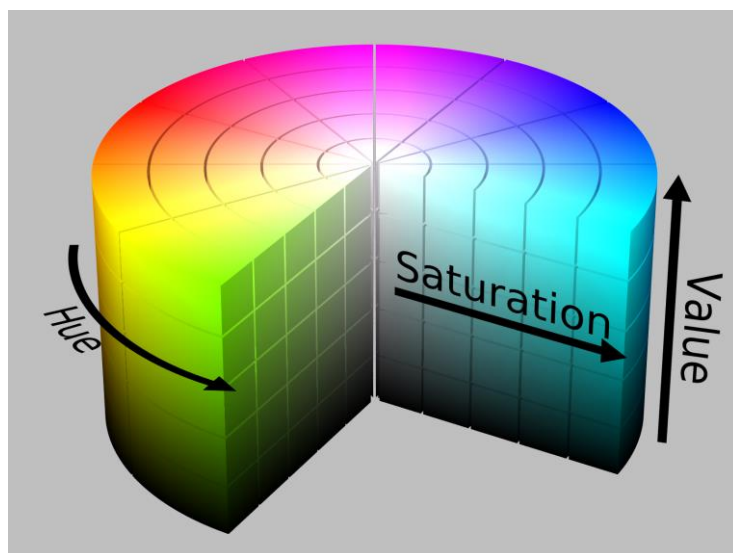


Рис.8 Представление цветового пространства HSV

Оттенок (H) изменяется по окружности конуса, насыщенность (S) цвета возрастает с отдалением от оси конуса, а яркость (V) – с приближением к его основанию. Различные оттенки и цвета определены при движении против часовой стрелки от 0 до 360 градусов. Насыщенность варьируется в пределах от 0 до 1 и представляет собой «чистоту цвета» – чем ближе этот параметр к 0, тем ближе цвет к нейтральному серому. V содержит информацию о яркости пикселя, принимает значения от 0 до 1. [21]

Таким образом, мы выбрали все элементы нашего алгоритма и можем приступить к его практической реализации.

3. Практическая реализация

3.1 Инструментарий разработки

Разработка системы велась на языке программирования C++ в среде разработки Microsoft Visual Studio 2013. C++ компилируемый язык программирования общего назначения, сочетает свойства как высокоуровневых, так и низкоуровневых языков программирования. В сравнении с его предшественником, языком программирования Си, наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования. Язык программирования C++ широко используется для разработки программного обеспечения. А именно, создание разнообразных прикладных программ, разработка операционных систем,

драйверов устройств, а также видео игр и многое другое. Существует несколько реализаций языка программирования C++ — как бесплатных, так и коммерческих. [22]

Немаловажным критерием в пользу выбора данного языка программирования для разработки является опыт разработки на нем, полученный за время обучения.

С целью упрощения процесса разработки было решено использовать библиотеку компьютерного зрения с открытым исходным кодом (Open Source Computer Vision Library) OpenCV. Она содержит более 500 функций, способных работать в реальном времени. Эта библиотека содержит алгоритмы для обработки изображений, распознавания образов, работы с видео, для слежения за объектами и др.

Изначально OpenCV создавалась для исследования машинного зрения, разработки и оптимизации кода. Приложения должны быть переносимыми, оптимизированными, код которых не обязательно должен быть открытым. Приложения также могут создаваться в коммерческих целях. Существуют версии для ОС Windows, Mac, Linux. Поддерживаются различные языки, в том числе C/C++, Python, Java. [23]

Библиотека состоит из нескольких модулей:

- ядро, которое содержит базовые структуры, матричную алгебру, алгоритмы работы с памятью, алгоритмы преобразования типов, алгоритмы для обработки ошибок, функции для работы с 2D графиками;
- модуль обработки изображений, он включает функции для работы с изображениями (преобразование, фильтрация и т.д.), функции для анализа изображений (поиск контуров, гистограммы и т.д.), алгоритмы анализа движений, слежение за объектами, алгоритмы распознавания объектов (лиц, предметов);
- машинное обучение, этот модуль содержит функции для классификации и анализа данных;
- модуль для создания пользовательского интерфейса, отвечающий за создание окон, вывод изображений, захват видео из файлов и камер, чтение/запись изображений;

- устаревшие функции, такие как пространственное зрение, нахождение и описание черт лица, поиск стереосоответствий, описание текстур.

Мало кто знает обо всех областях, где используется компьютерное зрение. Например, большинство знают, что оно используется в системах видеонаблюдения, применяется для обработки фото и видео в интернете, в аэрофотосъемке, в робототехнике. Но немногие знают, насколько обыкновенным делом стало машинное зрение в производстве: практически все, что выпускается серийно, на каком-то этапе автоматически проверяется с использованием компьютерного зрения.

Открытая лицензия для OpenCV была составлена таким образом, чтобы было возможно создавать коммерческие приложения, используя любые возможности OpenCV. Не обязательно делать проект открытым или публиковать изменения, внесенные при его создании в библиотеку. В большей степени благодаря этим условиям существует большое сообщество пользователей, включающее в себя такие крупные компании, как IBM, Microsoft, Intel, Sony, Google, и это далеко не полный список, а также научно-исследовательские центры, такие как Стэнфорд, Массачусетский технологический институт, CMU, Кембридж. OpenCV популярна во всем мире, причем большие сообщества пользователей можно найти в Китае, Японии, России, Европе и Израиле.

С момента своего первого релиза в январе 1999 года, OpenCV была использована во многих приложениях и научно-исследовательских работах, в том числе: наложение обычных карт и фотографий со спутника, выравнивание документов при сканировании, удаление шума из медицинских изображений, анализ объектов, системы безопасности, автоматическое наблюдение, системы контроля качества на производстве, калибровка камер, а также беспилотные летательные, наземные и подводные аппараты. Она даже была использована для распознавания звука и музыки, где методы распознавания образов были применены к изображениям спектрограмм звука.

Таким образом, библиотека OpenCV является достаточно универсальным инструментом для работы с изображениями и видео, поэтому было принято решение использовать именно ее для реализации работы.

3.2 Реализация алгоритма компоненты

В связи с тем, что существует огромное разнообразие дорожных знаков, перед тем, как реализовать программную компоненту, необходимо разобраться с теми знаками, которые будут распознаваться. К дорожным знакам, которые необходимо отразить в нашем алгоритме относится знак 3.24 «Ограничение максимальной скорости» из группы «Запрещающие знаки», знак 4.6 «Ограничение минимальной скорости» из группы «Предписывающие знаки» и знак 6.2 «Рекомендуемая скорость» из группы «Информационные знаки».

Но без труда можно добавить любые другие дорожные знаки, путем создания каскадного классификатора.



Рис. 9 Дорожные знаки, отраженные в нашем алгоритме

В ходе выполнения работы мы решаем задачу распознавания дорожных знаков в реальном времени, поэтому входными данными является последовательность кадров. На вход подается последовательность изображений, снятых с определенным шагом, и требуется определить наличие знака на изображении, и если он есть, то «прочитать» информацию и выдать ее на выходе работы программы в соответствии с расположением знака, то есть на выходе данные должны появляться в той последовательности, в которой поступали кадры.

Одной из проблем, связанной с формированием выходных данных не для одного кадра, а для целой последовательности, является возможное повторение одного знака на нескольких последовательных изображениях. Такая ситуация разрешается следующим образом:

1. Знак распознается на всех кадрах, где он может быть распознан;
2. Происходит сравнение результатов распознавания с результатами соседних кадров. Если результаты совпали (как минимум на 90%, так как распознавание

не всегда дает на 100% верный результат, то запоминается только один из результатов, второй игнорируется. Если результаты отличаются, то считается, что распознавались различные знаки, тогда сохраняются оба результата.

В связи с вышесказанным можно определить следующий алгоритм действий над входным изображением (с использованием функций обработки изображений библиотеки OpenCV):

1. Получаем последовательность кадров из видеоряда.
2. Выполняем преобразование кадра. Сначала кадр преобразуется из RGB в HSV (меняется цветовое пространство, при этом использование цветовых составляющих заменяется оперированием оттенком цвета и насыщенностью). Затем применяем детектор границ Кенни и фильтр Гаусса.
3. Определяем наличия знака в кадре: просматриваются все пиксели, и получившаяся область, подходящая под параметры искомого объекта (то есть залитая белым цветом), считается областью интересов – ROI (Region Of Interest – регион интересов – интересующая область изображения). ROI позволяет задать определённую прямоугольную область изображения, таким образом можно ее выделить и работать в дальнейшем только с ней, не затрагивая всё изображение. ROI можно использовать для вырезания части изображения или, наоборот, добавления изображения. В описываемой реализации ROI выделяется функцией `rectangle()` (обводится прямоугольником) и сохраняется в качестве исходного изображения для дальнейшей работы. По окончании текущего этапа распознавания на выходе получается интересующая область – полотно знака.
4. Применяем каскадный классификатор к полученной области и формируем результаты.
5. Проверяем на распознавание одного знака с соседних кадров (удаляем дубликаты).

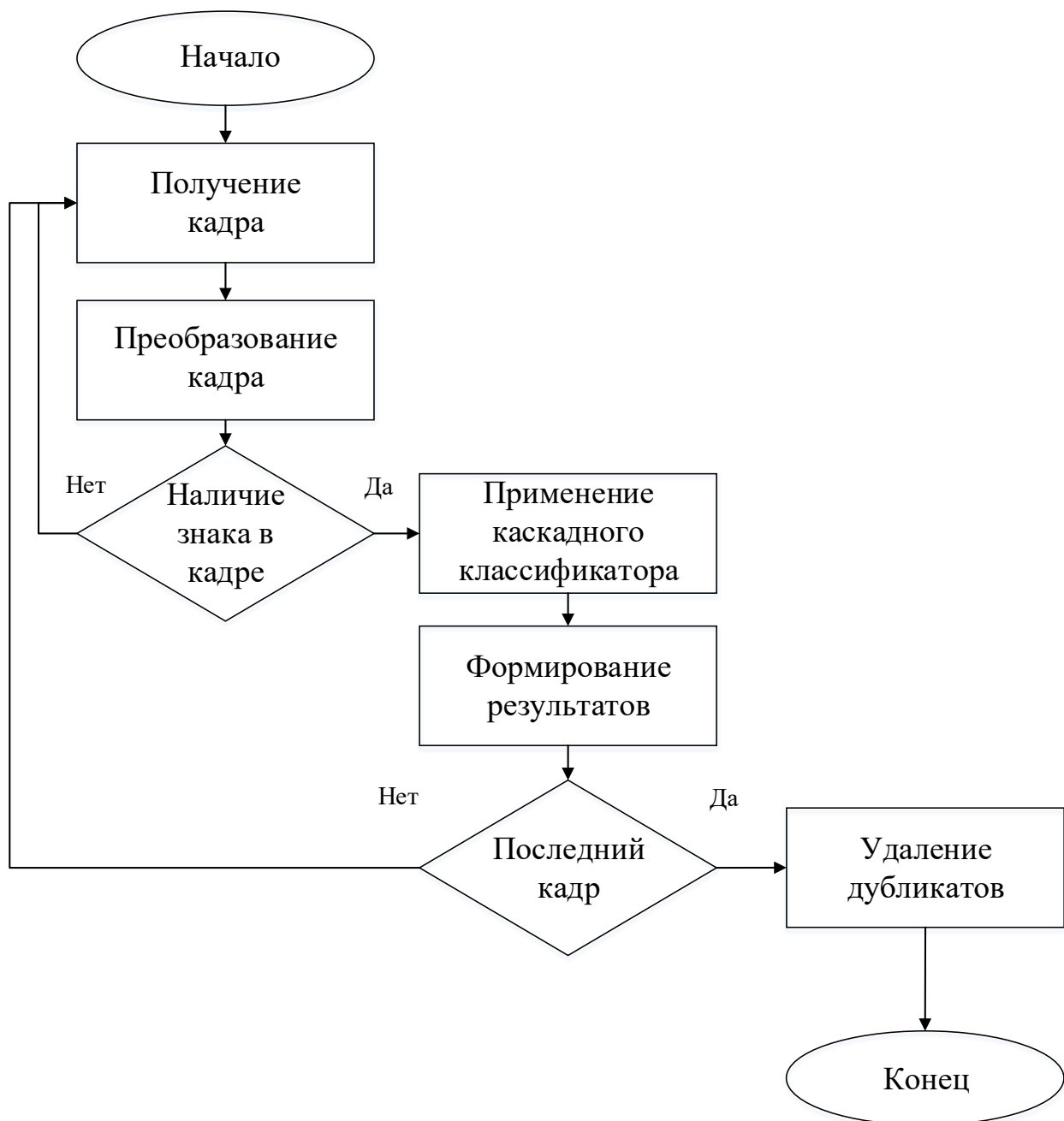


Рис. 10 Блок-схема алгоритма

3.3 Выделение элементов знака

На данном этапе ставится задача выделить отдельные символы на изображении знака. Для этого полученное ранее изображение преобразуется в представленное оттенками серого, то есть одноканальное. Далее оно размывается, чтобы убрать шумы, и бинаризуется. С помощью функции `threshold()`, порогом для которой задано значение 150 (подобранное опытным путем), выделяются регионы изображения, соответствующие объектам, которые необходимо проана-

лизовать. Это разделение основано на изменении интенсивности между пикселями объекта и пикселями фона. Чтобы отделить пиксели области интересов от фона, проводится сравнение значения интенсивности каждого пикселя по отношению к порогу. Далее пикселям устанавливаются значения в зависимости от результатов сравнения, те, значение которых было больше порогового, получили значение 255 (белый цвет), а остальные 0 (черный цвет).

3.4 Создание классификатора

В предыдущей главе мы подробно останавливались на теоретических основах каскадного классификатора, теперь перейдем к его практической реализации. Работа с каскадным классификатором включает два этапа: обучение и детектирование. В OpenCV есть два приложения для тренировки каскадов: `opencv_haartraining` и `opencv_traincascade` (новая версия, написанная на C++). Данные приложения хранят обученные классификаторы в различных форматах. Также для обучения используются другие утилиты:

1. `opencv_createsamples` используется для подготовки учебного набора положительных и тестовых образцов в формате, который поддерживается как `opencv_haartraining` так и `opencv_traincascade` приложениями. На выходе получается файл с расширением `*.vec`, это двоичный формат, который содержит изображение.

2. `opencv_performance` может быть использовано для оценки качества классификаторов, но только для обученных `opencv_haartraining`. Она использует коллекцию размеченных изображений, запускает классификатор и сообщает так называемую производительность, т.е. количество найденных объектов, количество пропущенных объектов, количество ложных срабатываний и другую информацию.

Для обучения необходимо собрать образы. Есть два типа образов: негативные и позитивные. Негативные образы соответствуют отсутствию объекта на изображении. Положительные образы соответствуют изображением с обнаруженными объектами. Набор негативных образов должен быть подготовлен вручную, в то время как множество положительных образов создается с помощью утилиты `opencv_createsamples`. [24]



Рис.11 Пример негативной выборки

Негативные образы должны быть взяты из произвольного видеоряда, причем они не должны содержать обнаруженных объектов. Они перечислены в специальном текстовом файле, в котором каждая строка содержит имя файла изображения (относительно каталога файла) с негативным образом. Пример описываемого файла:

Структура каталогов:

/images

image1.jpg

image2.jpg

test.txt

text.txt:

images/image1.jpg

images/image2.jpg

Позитивные образы создаются утилитой `opencv_createsamples`. В нашем случае, мы будем набирать выборку изображений дорожных знаков из видеоряда. Для корректной работы программы потребуется более тысячи изображений.



Рис. 12 Пример позитивной выборки

Количество и диапазон случайности можно менять с помощью аргументов командной строки утилиты `opencv_createsamples`

Аргументы:

-vec <vec_file_name>

Имя выходного файла, содержащего позитивные образы для обучения.

-img <image_file_name>

Источник изображения объекта.

-bg <background_file_name>

Файл описание фона, который содержит список изображений, которые используются в качестве фона для случайно искаженных версий объекта.

-num <number_of_samples>

Количество положительных образов для генерации.

-bgcolor <background_color>

Выбор цвета фона. Так как могут быть артефакты сжатия, количество цветов допуска может быть задано функцией `-bgthresh`. Все пиксели в диапазоне `[bgcolor-bgthresh, bgcolor+bgthresh]` интерпретируются как прозрачные.

`-bgthresh <background_color_threshold>`

`-inv`

Если указано, то цвета будут обращены.

`-randinv`

Если указано, то цвета будут обращены в случайном порядке.

`-maxidev <max_intensity_deviation>`

Максимальное отклонение интенсивностей пикселей на переднем плане образов.

`-maxxangle <max_x_rotation_angle>`

`-maxxangle <max_y_rotation_angle>`

`-maxxangle <max_z_rotation_angle>`

Максимальные углы поворота должны быть предоставлены в радианах.

`-show`

Полезные опции отладки. Если указано, то каждый образец будет показан. Нажатие Esc даст сигнал продолжать процесс создания образов без отображения.

`-w <sample_width>`

Ширина (в пикселях) выходных примеров.

`-h <sample_height>`

Высота (в пикселях) выходных примеров.

Следующая процедура используется для создания экземпляра образа объекта: исходное изображение поворачивается случайно вокруг всех трех осей. Выбранный угол ограничен максимумами угла. Пиксели, имеющие интенсивность

в диапазоне $[bg_color - bg_color_threshold; bg_color + bg_color_threshold]$ интерпретируются как прозрачные. Белый шум добавляется к интенсивности на переднем плане. Если *-inv* ключ указан, то пиксели на переднем плане инвертируются. Если *-randinv* ключ указан, то алгоритм случайным образом выбирает, будет ли обращение пикселей применено к этому образцу. Наконец, полученное изображение помещается на произвольный фон из файла описания фона, масштабировать до нужного размера, указанного в *-h* и *-w* и сохраняется в *vec*-файл, заданный в параметре *-vec*. [25]

Позитивные примеры также могут быть получены из коллекции ранее размеченных изображений. Эта коллекция, которая описывается в текстовом файле, похожа на файл описания фона. Каждая строка этого файла соответствует изображению. Первым элементом линии является имя файла. За ней следуют количество экземпляров объекта. Следующие числа являются координатами объектов ограничивающих прямоугольников (x, y, ширина, высота).

Структура каталогов:

/images

image1.jpg

image2.jpg

info.dat

info.dat:

images/image1.jpg 1 140 100 45 45

images/image2.jpg 2 100 200 50 50 50 30 25 25

Т.е. первое изображение дает одно положение объекта, а второе два. Для того, чтобы создать позитивные образы из таких коллекций, *-info* аргумент должен быть указан вместо *-img*:

-info <collection_file_name>

Схема по созданию образов в данном случае заключается в следующем. Экземпляры объектов взяты из изображений. Затем они изменяются до размера

целевых образов и сохраняются в выходной *vec*-файл. Никаких искажений не применяется, поэтому влияющие аргументы это: *-w*, *-h*, *-show* и *-Num*. *opencv_createsamples* может быть использована для изучения образов, хранящихся в файле позитивных образов. Для того, чтобы сделать это только *-vec*, *-w* и *-h* параметры должны быть указаны.

Следующим шагом является подготовка классификатора. Аргументы командной строки *opencv_traincascade* приложения сгруппированы по целям:

1. Общие аргументы:

-data <cascade_dir_name>

Где обученный классификатор должен быть сохранен.

-vec <vec_file_name>

vec-файл с позитивными образами (созданный утилитой *opencv_createsamples*).

-bg <background_file_name>

Файл описания фона.

-numPos <number_of_positive_samples>

-numNeg <number_of_negative_samples>

Количество позитивных / негативных образов, используемых в обучении для каждого этапа классификатора.

-numStages <number_of_stages>

Количество этапов каскада для обучения.

-precalcValBufSize <precalculated_vals_buffer_size_in_Mb>

Размер буфера — расчетное значение функции (в Мб).

-precalcIdxBufSize <precalculated_idx_buffer_size_in_Mb>

Размер буфера для индексов расчетных функций (в Мб). Чем больше памяти у вас есть, тем быстрее процесс обучения.

-baseFormatSave

Этот аргумент является актуальным в случае Хаар-подобных функций. Если он указан, то каскад будет сохранен в старом формате.

2. Параметры каскада:

-stageType <BOOST(default)>

Тип этапов. Только boosted классификаторы поддерживаются как этап типа на данный момент.

-featureType<{HAAR(default), LBP}>

Тип функции: HAAR — Хаара-подобные функции, LBP — местные модели двоичные.

-w <.sampleWidth >

-h <.sampleHeight >

Размер обучаемых образов (в пикселях). Должно быть ровно те же значения, которые используются во время создания образов обучения (opencv_createsamples).

3. Параметры Boosted-классификатора:

-bt <{DAB, RAB, LB, GAB(default)}>

Тип boosted классификаторов: DAB — Дискретная AdaBoost, RAB — Real AdaBoost, LB — LogitBoost, GAB — Gentle AdaBoost.

-minHitRate <min_hit_rate>

Минимальный желаемый показатель попадания на каждом этапе классификатора.

-maxFalseAlarmRate <max_false_alarm_rate>

Максимальный желаемый уровень ложных тревог для каждого этапа классификатора.

-weightTrimRate <weight_trim_rate>

Указывает, будет ли использована обрезка и ее вес. Достойный выбор 0,95.

-maxDepth <max_depth_of_weak_tree>

Максимальная глубина слабого дерева. Достойный выбор 1.

-maxWeakCount <max_weak_tree_count>

Максимальное количество слабых деревьев для каждого этапа каскада. Boosted классификатор имеет так много слабых деревьев ($\leq \text{maxWeakCount}$), сколько необходимо для достижения заданного maxFalseAlarmRate .

4. Параметры для особенностей Хаара:

-mode <BASIC (default) / CORE / ALL>

Полученный каскадный классификатор представлен в приложении А.

3.5 Достигнутые результаты

Для проверки корректности работы программы и получения статистических данных были подготовлены реальные данные, представляющие собой последовательность кадров, снятых в г. Томске с помощью видеорегистратора. Размеры кадров – 1920x1080 пикселей, изображения четкие.

Путем анализа большого числа изображений, содержащих знаки, было экспериментально установлено, что для достижения поставленной цели – распознавания информационных знаков – целесообразным выглядит ограничение размеров охватывающего прямоугольника. Значения H , S , V зависят от условий освещенности и могут корректироваться пользователем программы, например, для данных погодных условий. Для этого конкретного примера были выбраны диапазоны значений 100 – 200, 210 – 255, 0 – 255 для H , S и V соответственно. Результат обработки последовательности кадров представлен на рисунке.

Для того, чтобы сделать определённые выводы о точности распознавания, была проведена исследовательская работа. В первой главе мы проводили обзор имеющихся решений и для проверки работы нашего алгоритма необходимо сравнить несколько методов между собой. Для сравнения возьмем следующие методы: распознавание по цвету, преобразование Хафа, каскадный классификатор без обработки изображения и наш двухступенчатый алгоритм.

Для более правильной оценки качества алгоритма воспользуемся $f1$ -метрикой, а не стандартным отношением числа правильно распознанных дорожных знаков к их общему числу. Это связано с тем, что стандартная точность алгоритма присваивает всем документам одинаковый вес, что может быть некорректно в случае если распределение документов в обучающей выборке сильно смещено в сторону какого-то одного или нескольких классов. В этом случае у классификатора есть больше информации по этим классам и соответственно в рамках этих классов он будет принимать более адекватные решения.

На практике это приводит к тому, что мы можем иметь точность в 80%, но при этом в рамках какого-то конкретного класса классификатор может работать, плохо, не определяя правильно даже треть документов. Один выход из этой ситуации заключается в том, чтобы обучать классификатор на специально подготовленном, сбалансированном корпусе документов. Минус этого решения в том, что мы отбираем у классификатора информацию об относительной частоте документов. Эта информация при прочих равных может оказаться очень кстати для принятия правильного решения.

Другой выход заключается в изменении подхода к формальной оценке качества. Точность (precision) и полнота (recall) являются метриками которые используются при оценке большей части алгоритмов извлечения информации. Иногда они используются сами по себе, иногда в качестве базиса для производных метрик, таких как F-мера или R-Precision.

Точность системы в пределах класса – это доля документов, действительно принадлежащих данному классу относительно всех документов, которые система отнесла к этому классу. Полнота системы – это доля найденных классификатором документов, принадлежащих классу относительно всех документов этого класса в тестовой выборке. Показатели точности и полноты для выбранных алгоритмов представлены в Таблице 1.

Табл. 1 Показатели точности и полноты для выбранных алгоритмов

Параметры	Распознавание по цвету	Преобразование Хафа	Каскадный классификатор без обработки изображения	Двухступенча- тый алгоритм
Точность	25,5	31,7	33,2	54,1
Полнота	45,6	63,9	72,1	87,1

Исходя из данной таблицы, можно сделать вывод о применимости нашего алгоритма. Наихудший результат показал алгоритм, основанный на распознавании по цвету. Это неудивительно, т.к. в сложных погодных условиях дорожная картина значительно искажается, что влияет на точность распознавания. Примерно равные результаты показали преобразование Хафа и каскадный классификатор без обработки изображения. Наилучший же результат как по точности, так и по полноте показал наш разработанный двухступенчатый алгоритм на основе каскадного классификатора с применением предобработки изображения. Примеры детектирования изображений представлены на рис. 13.

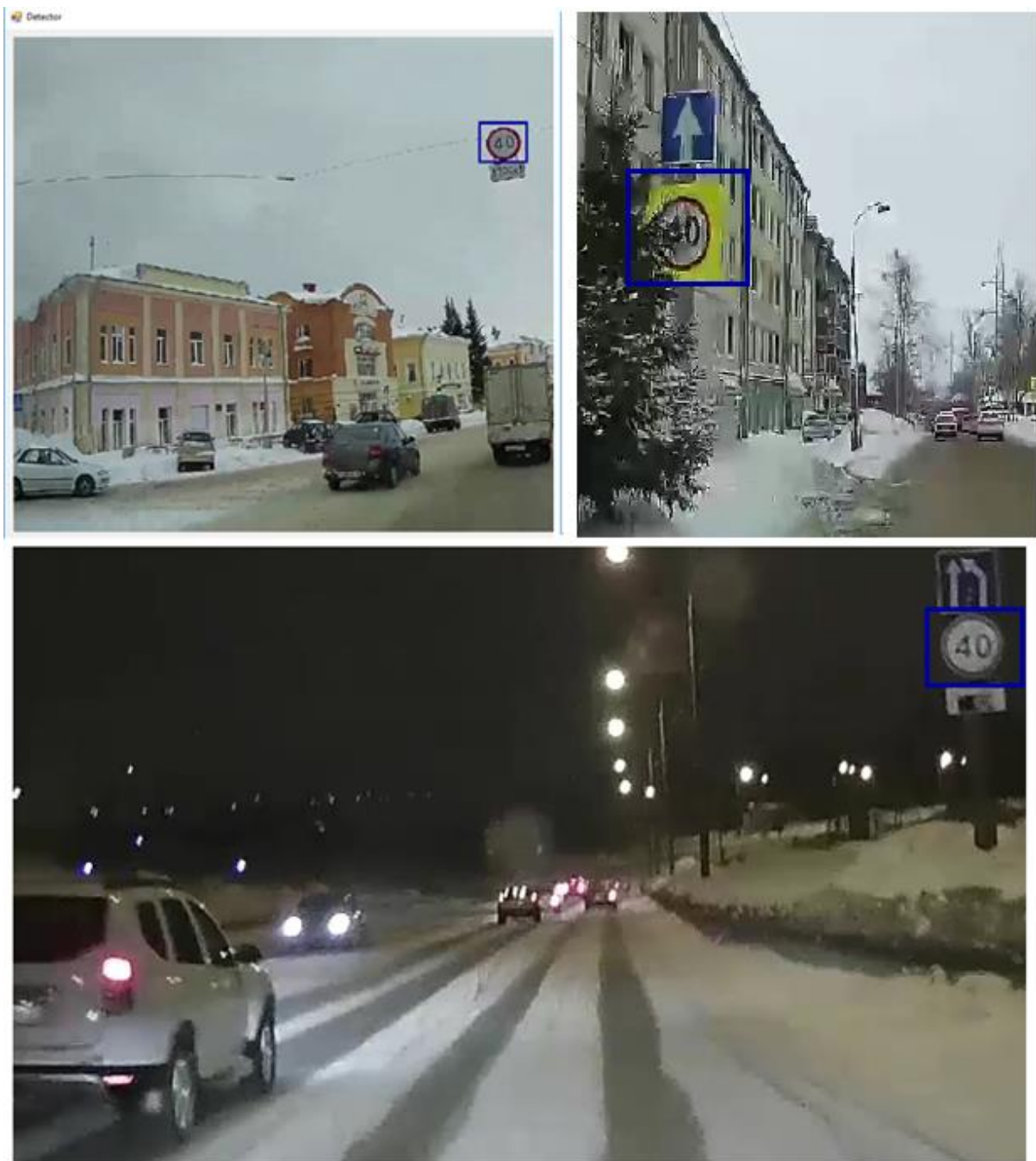


Рис. 13 Пример детектирования изображений

Кроме того, необходимо сделать тест на возможность распознавания дорожных знаков в реальном времени. После обучения программа выдаёт обученный каскад в виде xml файла, который может использоваться непосредственно для детектирования объектов. Чтобы протестировать его, мы снова генерируем 1000 объектов по принципу, создавая тем самым проверочную выборку.

Для тестирования каскада используется модуль performance. Загружаем в него проверочную выборку и каскад и через некоторое время мы можем наблюдать в консоли такую картину:

File Name	Hits	Missed	False
0000_0000_0000_0000_0001.jpg	1	0	0
0000_0000_0000_0000_0002.jpg	1	0	1
0000_0000_0000_0000_0003.jpg	0	1	0
0000_0000_0000_0000_0004.jpg	1	0	0
.....			
Total	339	611	532

Number of stages: 15
 Number of weak classifiers: 89
 Total time: 16.334000

Обратим внимание на пункт «Total time». Это время, которое понадобилось для обработки 1000 изображений. С учётом того, что их нужно было считать с диска, время, затраченное на обработку одного изображения, составляет доли секунды:

$$t = \frac{16,334}{1000} = 16,334 \text{ мс}$$

С учетом того, что мы решаем задачу распознавания дорожных знаков в реальном времени, то посчитаем, будет ли достаточно такой производительности для работы в реальном времени. Стандартный видеоряд состоит из 24 кадров в секунду: исходя из этого, мы можем посчитать за какое время эти кадры могут быть обработаны:

$$t = 16,334 \cdot 24 = 392 \text{ мс} = 0,392 \text{ с}$$

Полученный результат говорит о том, что задача распознавания дорожных знаков в реальном времени может быть успешно решена с помощью данного алгоритма.

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8Е31	Кремлев Иван Андреевич

Институт	Институт кибернетики	Кафедра	СУМ
Уровень образования	Бакалавриат	Направление/специальность	15.03.06 Мехатроника и робототехника

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	- Тариф на электроэнергию – 5,257 руб./кВт·ч.; - Оклад студента – 1850 руб. в месяц; - Оклад руководителя проекта – 13824 руб. в месяц.
2. Нормы и нормативы расходования ресурсов	- Годовая норма амортизации составляет 40 %
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	На основании пункта 1 ст.58 закона №212-ФЗ для учреждений осуществляющих образовательную и научную деятельность вводится пониженная ставка – 27,1%.

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	- Методы коммерциализации результатов инженерных решений; - SWOT-анализ.
2. Планирование и формирование бюджета научных исследований	- Определение трудоемкости выполнения работ; - Расчет материальных затрат НИИ; - Основная и дополнительная зарплата исполнителей темы; - Отчисления во внебюджетные фонды; - Накладные расходы; - Проведение анализа безубыточности проекта
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	- Расчет интегрального показателя финансовой эффективности.

Перечень графического материала (с точным указанием обязательных чертежей):

1. Матрица SWOT
2. График проведения и бюджет НИ
3. График безубыточности проекта
4. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику	18.03.2017
--	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. МЕН	Тухватулина Л.Р.	к.ф.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8Е31	Кремлев Иван Андреевич		

4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

4.1 Оценка коммерческого потенциала, перспективности и альтернатив проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

4.1.1 Методы коммерциализации результатов инженерных решений

При реализации инженерного решения одним из важнейших компонентов является коммерциализация. Владелец соответствующих объектов интеллектуальной собственности может преследовать различные цели, которые во многом зависят от того, куда в последующем будет направлен полученный коммерческий эффект. В данном проекте целью является получение средств для продолжения своих научных исследований и разработок (получение финансирования, оборудования, уникальных материалов, других научно-технических разработок и т.д.), а также обеспечение постоянного притока финансовых средств. Из методов коммерциализации наиболее подходящим будет являться торговля патентными лицензиями, т.е. передача третьим лицам права использования объектов интеллектуальной собственности на лицензионной основе, а именно алгоритма программной компоненты. Данный метод был выбран исходя из возможности получения наибольшей экономической эффективности. За продажу лицензий выплачивается вознаграждение (лицензионные платежи), а именно:

- **периодические отчисления (роялти)** от дохода покупателя в течение всего периода действия соглашения, позволяющие установить прямую зависимость размера вознаграждения от экономического эффекта использования предмета лицензии;
- **единовременные платежи**, устанавливаемые заранее на основе экспертных оценок. Эти платежи не связаны во времени с фактическим использованием лицензии.

Уровень роялти варьирует в пределах от 2 до 10%, но наиболее часто встречаются ставки в 3—5%. Единовременные платежи осуществляются в форме паушальных платежей, а также посредством передачи части ценных бумаг и в дру-

гих формах. Паушальные платежи — это твердо зафиксированная в лицензионном соглашении сумма вознаграждения, которая выплачивается один раз и размер которой не зависит от объема производства или сбыта лицензионной продукции. [26]

Исходя из текущей стадии проекта, данный метод коммерциализации будет являться наиболее подходящим, т.к. он с одной стороны позволяет получать паушальные платежи, позволяющие продолжать научные исследования и разработки и улучшать уже имеющиеся алгоритмы. С другой стороны, роялти позволяют получить приток финансовых средств, который в дальнейшем может позволить открыть собственную организацию.

4.1.2 SWOT-анализ

Для исследования внешней и внутренней среды проекта, проведем SWOT-анализ, который состоит из определения сильных, слабых сторон проекта, его возможностей и угроз (таблица 2).

Табл. 2 Матрица SWOT-анализа

	<p>Сильные стороны научно-исследовательского проекта:</p> <p>С1. Универсальность системы.</p> <p>С2. Низкая стоимость производства</p> <p>С3. Простая интеграция системы в автомобиль</p> <p>С4. Простота эксплуатации</p>	<p>Слабые стороны научно-исследовательского проекта:</p> <p>Сл.1 Сложность тестирования в реальных условиях</p> <p>Сл2. Нет возможности использовать компоненту на ОС, отличной от Windows OS</p> <p>Сл3. Серьезные затраты на интеграцию проекта</p>
<p>Возможности:</p> <p>В1. Развитие беспилотных автомобилей</p> <p>В2. Развитие системы на базе НИ ТПУ</p> <p>В3. Появление новых, более совершенных</p>	<p>Развитие беспилотных автомобилей, а также появление новых технологий приведет к снижению стоимости готового продукта для потребителя и более простой эксплуатации автомобиля.</p>	<p>Тестирование программной компоненты в реальных условиях будет затруднительно в связи с несовершенством имеющихся на сегодняшний день технологий.</p>

технологий		
В4. Выпуск специалистов в данной отрасли		
Угрозы: У1. Низкая развитость робототехники в России У2. Более совершенные аналоги конкурентов У3. Несовершенство в законодательстве У4. Незаинтересованность в разработке	Низкая развитость робототехники в России, несовершенство в российском законодательстве и незаинтересованность в разработке со стороны крупных компаний может стать серьезным препятствием в развитии проекта.	Для интеграции проекта потребуются серьезные средства, но финансирование со стороны крупных компаний, придаст проекту возможность для развития и распространения технологий в России.

Для более четкого понимания взаимосвязей в таблице SWOT-анализ реализуем интерактивные матрицы проектов.

Табл. 3 Интерактивная матрица сильных сторон и возможностей проекта

		C1	C2	C3	C4
Возможности проекта	B1	-	+	+	+
	B2	+	-	-	-
	B3	-	+	-	-
	B4	-	-	-	+

Интерактивные матрицы сильных сторон и угроз, слабых сторон и возможностей, слабых сторон и угроз выполнены по аналогии (представлены в Приложении В).

4.2 Планирование и формирование бюджета научных исследований

4.2.1 Планирование научно-исследовательских работ

Планирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках научного исследования;
- определение участников каждой работы;
- установление продолжительности работ;

- построение графика проведения научных исследований.

Для выполнения научных исследований формируется рабочая группа, в состав которой могут входить научные сотрудники и преподаватели, инженеры, техники и лаборанты, численность групп может варьироваться. По каждому виду запланированных работ устанавливается соответствующая должность исполнителей. Рабочая группа, выполняющая научные исследования, состоит из двух человек: научного руководителя и непосредственного исполнителя – студента. [27]

Табл. 4 План научно-исследовательской работы

Основные этапы	№ этапа	Содержание работ	Должность исполнителя
Выбор направления исследования	1	Подбор и изучение материалов по теме	Руководитель, студент
	2	Проведение патентных исследований	Студент
	3	Выбор направления исследования	Руководитель, студент
Теоретические и экспериментальные исследования	4	Проведение теоретических расчетов и обоснований	Студент
	5	Разработка алгоритма программы	Студент
Разработка технической документации и практические исследования	6	Разработка блок-схемы	Студент
	7	Написание программного обеспечения	Студент
	8	Тестирование программного обеспечения в реальных условиях	Студент
Оформление результатов исследований	9	Анализ полученных данных	Руководитель, студент
	10	Составление отчета о полученной работе	Студент

4.2.2 Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаях образуют основную часть сто-

имости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования. Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Для определения ожидаемого значения трудоемкости $t_{\text{ожи}}$ используется следующая формула:

$$t_{\text{ожи}} = \frac{3t_{\text{min } i} + 2t_{\text{max } i}}{5}$$

где $t_{\text{ожи}}$ - ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{\text{min } i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;

$t_{\text{max } i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_{pi} , учитывающая параллельность выполнения работ несколькими исполнителями:

$$T_{pi} = \frac{t_{\text{ожи } i}}{\text{Ч}_i}$$

где T_{pi} – продолжительность одной работы, раб. дн.;

$t_{\text{ожи } i}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.

Ч_i – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

4.2.3 Разработка графика проведения научного исследования

Для удобства построения графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{pi} – продолжительность выполнения i -й работы в рабочих днях;

$k_{\text{кал}}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}}$$

где $T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году.

В 2017 году 365 дней; из них 118 выходных и праздничных дней. Коэффициент календарности рассчитаем следующим образом:

$$k_{\text{кал}} = \frac{365}{365 - 118} = 1,48$$

Рассчитанные значения в календарных днях по каждой работе необходимо округлить до целого числа. Результаты расчетов сведем в таблицу.

Название работы	Трудоёмкость работ						Исполнители	Длительность работ в рабо- чих днях T_{pi}		Длительность работ в ка- лендарных днях $T_{\kappa i}$	
	t_{\min} , чел-дни		t_{\max} , чел-дни		$t_{ожг}$, чел-дни						
	НР	С	НР	С	НР	С		НР – научный руко- водитель; С - студент	НР	С	НР
1. Подбор и изучение материалов по теме	10	10	20	20	14	14	НР, С	7	7	10	10
2. Проведение патентных исследований	-	7	-	11	-	8,6	С	-	8,6	-	13
3. Выбор направления исследований	4	4	8	8	5,6	5,6	НР, С	2,8	2,8	4	4
4. Проведение теоретических расчетов и обоснований	–	8	–	12	–	9,6	С	–	9,6	–	14
5. Разработка алгоритма программы	–	25	–	35	–	29	С	–	29	–	43

6. Разработка блок-схемы	-	2	-	5	-	3,2	С	-	3,2	-	5
7. Написание программного обеспечения	—	40	—	55	—	46	С	—	46	—	69
8. Тестирование программного обеспечения в реальных условиях	—	8	—	15	—	10,8	С	—	10,8	—	16
9. Анализ полученных результатов	8	8	14	14	10,4	10,4	НР, С	5,2	5,2	8	8
10. Составление отчета по проделанной работе	—	10	—	20	—	14	С	—	14	—	21
Итого								15	136,2	22	203

	Название задачи ▼	Длительность ▼	Начало ▼	Окончание ▼
1	Подбор и изучение материалов по теме	20 дней	Пн 03.10.16	Пт 28.10.16
2	Проведение патентных исследований	11 дней	Ср 05.10.16	Ср 19.10.16
3	Выбор направления исследования	8 дней	Пт 07.10.16	Вт 18.10.16
4	Проведение теоретических расчетов и обоснований	12 дней	Вт 18.10.16	Ср 02.11.16
5	Разработка алгоритма программы	35 дней	Чт 03.11.16	Ср 21.12.16
6	Разработка блок-схемы	5 дней	Чт 22.12.16	Ср 28.12.16
7	Написание программного обеспечения	55 дней	Чт 29.12.16	Ср 15.03.17
8	Тестирование программного обеспечения в реальных условиях	15 дней	Чт 16.03.17	Ср 05.04.17
9	Анализ полученных результатов	14 дней	Чт 06.04.17	Вт 25.04.17
10	Составление отчета о проделанной работе	20 дней	Ср 26.04.17	Вт 23.05.17

Рис. 14 Данные к диаграмме Ганта

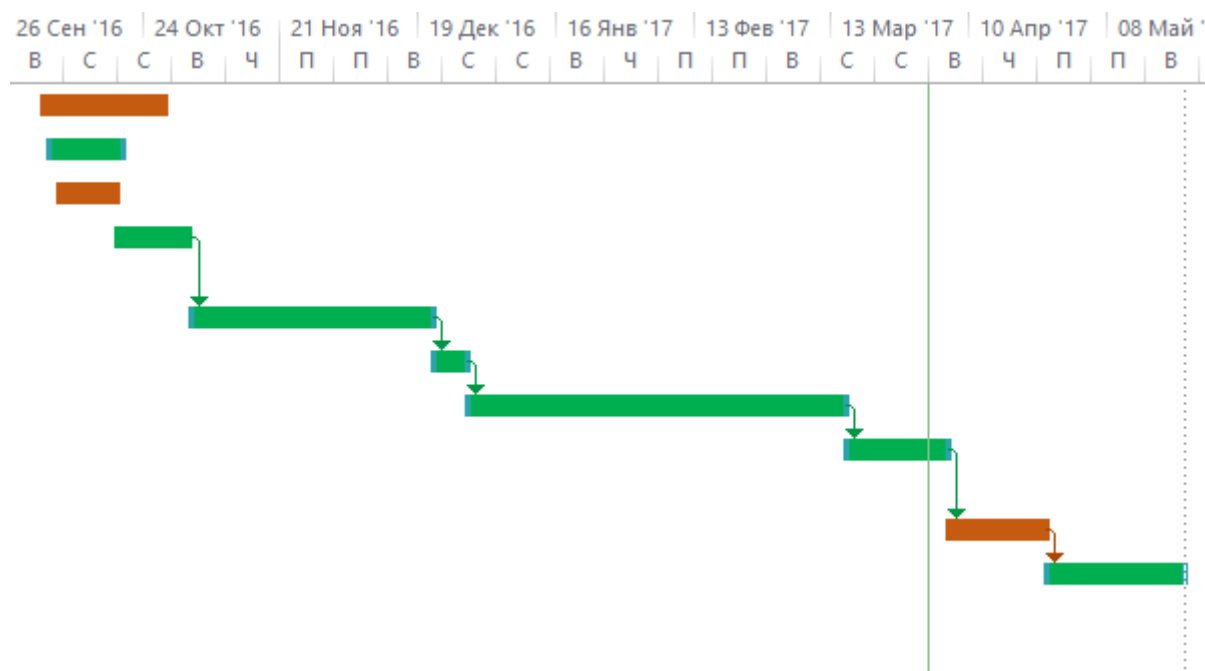


Рис. 15 Диаграмма Ганта

4.2.4 Расчет материальных затрат НТИ

Данная статья включает стоимость всех материалов, используемых при разработке проекта. В нашем случае мы будем учитывать только затраты на электроэнергию, потраченную в ходе выполнения проекта на работу используемого оборудования, рассчитываемые по формуле:

$$C_{\text{эл}} = P_{\text{об}} \cdot t_{\text{об}} \cdot C_{\text{э}}$$

где $P_{\text{об}}$ – мощность, потребляемая оборудованием, кВт;

$\text{Ц}_э$ – тариф на 1 кВт·час;

$t_{об}$ – время работы оборудования, час.

Для ТПУ $\text{Ц}_э = 5,257$ руб./кВт·час (с НДС). Время работы оборудования вычисляется на основе итоговых данных таблицы 2 для инженера (ТРД) из расчета, что продолжительность рабочего дня равна 8 часов.

$$t_{об} = T_{рД} \cdot K_t$$

где $K_t \leq 1$ – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к $T_{рД}$, определяется исполнителем самостоятельно. В ряде случаев возможно определение $t_{об}$ путем прямого учета, особенно при ограниченном использовании соответствующего оборудования. Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{об} = P_{ном} \cdot K_c$$

где $P_{ном}$ – номинальная мощность оборудования, кВт;

$K_c \leq 1$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности.

Для технологического оборудования малой мощности $K_c = 1$.

Пример расчета затраты на электроэнергию для технологических целей приведен в таблице 5.

Табл. 5 Расчет материальных затрат НТИ

Наименование оборудования	Время работы оборудования, $t_{об}$, час	Потребляемая мощность, $P_{об}$, кВт	Затраты на электроэнергию, $C_{эл}$, руб
Персональный компьютер	1088	0,4	2287,85
Итого:			2287,55

4.2.5 Расчет затрат на основные средства

Данная статья включает стоимость всех необходимых устройств, без которых невозможна реализация проекта. В нашем случае мы будем учитывать затраты на ноутбук и веб-камеру. Расчет материальных затрат осуществляется по следующей формуле:

$$З_M = (1 + k_T) \cdot \sum_{i=1}^m Ц_i \cdot N_{расхi}$$

где m – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расхi}$ – количество материальных ресурсов i -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, и т.д.);

$Ц_i$ – цена приобретения единицы i -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м² и т.д.);

k_T – коэффициент, учитывающий транспортно-заготовительные расходы.

Табл. 6 Расчет затрат на основные средства

Наименование	Ед. изм.	Количество		Цена за ед., руб.		Затраты на материалы, (З _М), руб.	
		Исп. 1	Исп. 2	Исп. 1	Исп. 2	Исп. 1	Исп. 2
Ноутбук	шт.	1	1	30000	20000	30000	20000
Веб-камера	шт.	1	1	1120	580	1120	580
Итого						31120	20580

4.2.6 Основная заработная плата исполнителей темы

В настоящую статью включается основная заработная плата научных и инженерно-технических работников, рабочих макетных мастерских и опытных производств, непосредственно участвующих в выполнении работ по данной теме. Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы окладов и тарифных ставок.

В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20 –30 % от тарифа или оклада.

Статья включает основную заработную плату работников, непосредственно занятых выполнением НТИ, (включая премии, доплаты) и дополнительную заработную плату:

$$З_{зп} = З_{осн} + З_{доп}$$

где $З_{осн}$ – основная заработная плата;

$З_{доп}$ – дополнительная заработная плата (12-20 % от $З_{осн}$).

Основная заработная плата ($З_{осн}$) руководителя (лаборанта, инженера) от предприятия (при наличии руководителя от предприятия) рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \cdot T_p$$

где $З_{осн}$ – основная заработная плата одного работника;

T_p – продолжительность работ, выполняемых научно-техническим работником, раб. дн. (табл.);

$З_{дн}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_M \cdot M}{F_D}$$

где $З_M$ – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года: при отпуске в 24 раб. дня $M=11,2$ месяца, 5-дневная неделя; при отпуске в 48 раб. дней $M=10,4$ месяца, 6-дневная неделя;

F_D – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн. (Таблица 7)

Табл.7 Баланс рабочего времени

Показатели рабочего времени	Руководитель	Студент
Календарное число дней	365	365
Количество нерабочих дней - выходные дни	120	120

- праздничные дни		
Потери рабочего времени - отпуск - невыходы по болезни	50	70
Действительный годовой фонд рабочего времени	195	175

Месячный должностной оклад работника рассчитывается по формуле:

$$З_M = З_{TC} \cdot (1 + k_{пр} + k_D) \cdot k_p$$

где $З_{TC}$ – заработная плата по тарифной ставке, руб.;

$k_{пр}$ – премиальный коэффициент, равный 0,3 (т.е. 30% от $З_{TC}$);

k_D – коэффициент доплат и надбавок составляет примерно 0,2 – 0,5 (в НИИ и на промышленных предприятиях – за расширение сфер обслуживания, за профессиональное мастерство, за вредные условия: 15-20 % от $З_{TC}$);

k_p – районный коэффициент, равный 1,3 (для Томска).

Расчет основной заработной платы представлен в Таблице 8.

Табл. 8 Основная заработная плата исполнителей темы

Исполнитель	Оклад (руб.)	Среднедневная заработная плата за день (руб./дн.)	Трудоемкость(дн.)		Основная зарплата (руб.)	
			Исп. 1	Исп. 2	Исп. 1	Исп. 2
Научный руководитель	13000	634,15	22	22	13951,30	13951,30
Студент	1850	90,24	203	151	18318,72	13626,24
Итого:					32270,02	27577,54

4.2.7 Дополнительная заработная плата исполнителей темы

Затраты по дополнительной заработной плате исполнителей темы учитывают величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.). Расчет дополнительной заработной платы ведется по следующей формуле:

$$З_{\text{доп}} = k_{\text{доп}} \cdot З_{\text{осн}}$$

где $k_{\text{доп}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12).

Табл. 9 Дополнительная заработная плата исполнителей темы

Исполнитель	Основная заработная плата (руб.)		Дополнительная заработная плата (руб.)	
	Исп. 1	Исп. 2	Исп. 1	Исп. 2
Научный руководитель	13951,30	13951,30	1674,16	1674,16
Студент	18318,72	13626,24	2198,25	1635,15
Итого			3872,41	3309,31

4.2.8 Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников. Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{\text{внеб}} = k_{\text{внеб}} \cdot (З_{\text{осн}} + З_{\text{доп}})$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.). На 2017 г. в соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30%. На основании пункта 1 ст.58 закона №212-ФЗ для учреждений, осуществляющих образовательную и научную деятельность, вводится пониженная ставка – 27,1%. На стипендию студента отчисления во внебюджетные фонды не осуществляются. Результаты расчетов представлены в таблице (табл. 10).

Табл. 10 Отчисления во внебюджетные фонды

Зарплата	Исполнения	
	Исп. 1	Исп. 2
Основная	13951,30	13951,30
Дополнительная	1674,16	1674,16
Коэффициент	0,271	
Итого	4234,50	4234,50

4.2.9 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$З_{\text{накл}} = (\text{сумма статей } 1 \div 5) \cdot k_{\text{нр}}$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов примем в размере 40%.

Табл. 11 Накладные расходы

Статья расходов	Исполнение	
	Исп. 1	Исп. 2
Материальные затраты НТИ	2287,85	2287,85

Затраты на основные средства	31120	20580
Основная зарплата	32270,02	27577,54
Дополнительная зарплата	3872,41	3309,31
Отчисления во внебюджетные фонды	4234,50	4234,50
Коэффициент накладных расходов	0,40	
Накладные расходы	29513,91	23195,68

4.2.10 Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно-исследовательской работы (темы) является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции. Определение бюджета затрат на научно-исследовательский проект по каждому варианту исполнения приведен в Табл. 12.

Табл. 12 Бюджет затрат научно-исследовательского проекта

Наименование статьи	Сумма, руб.		Примечание
	Исп.1	Исп.2	
1. Материальные затраты НТИ	2287,85	2287,85	Пункт 2.4.
2. Затраты на основные средства	31120	20580	Пункт 2.5.
3. Затраты по основной заработной плате исполнителей темы	32270,02	27577,54	Пункт 2.6.
4. Затраты по дополнительной заработной плате исполнителей темы	3872,41	3309,31	Пункт 2.7.

5. Отчисления во внебюджетные фонды	4234,50	4234,50	Пункт 2.8.
6. Накладные расходы	29513,91	23195,68	40 % от суммы Пунктов 2.4-2.8.
7. Бюджет затрат НТИ	103298,69	81184,88	Сумма пунктов 2.4.-2.9.

4.2.11 Проведение анализа безубыточности проекта

Точка безубыточности – минимальный объем производства и реализации продукции, при котором расходы будут компенсированы доходами, а при производстве и реализации каждой последующей единицы продукции предприятие начинает получать прибыль. Точку безубыточности можно определить в единицах продукции, в денежном выражении или с учётом ожидаемого размера прибыли. [28] В данной работе точку безубыточности будем находить графически. При расчете точки безубыточности исходят из принципа линейного наращивания объемов производства и продаж без учета возможностей скачка, например, вследствие сезонности выпуска и сбыта. График безубыточности проекта представлен на Рис. 16.

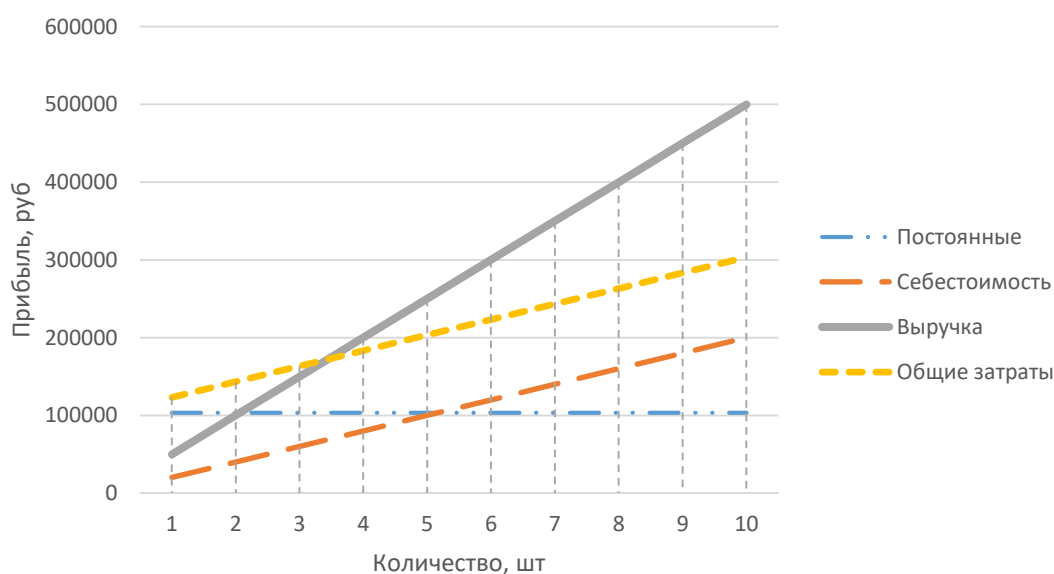


Рис. 16 График безубыточности проекта

Из графика видно, что при продаже более 3 лицензий на программную компоненту, проект достигнет точки безубыточности. Данный показатель говорит о серьезном коммерческом потенциале данной разработки.

4.3 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

4.3.1 Определение ресурсной, финансовой и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности. Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат трех (или более) вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принимается за базу расчета (как знаменатель), с которым соотносятся финансовые значения по всем вариантам исполнения. Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп } i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}$$

где $I_{\text{финр}}^{\text{исп } i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Табл. 13 Расчет интегрального финансового показателя

Исполнение	Бюджет НТИ, руб.	Интегральный финансовый показатель
Исполнение 1	103298,69	1
Исполнение 2	81184,88	0,79

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат разработки в разах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в разах. Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i$$

где I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i – бальная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания.

Табл. 14 Сравнительная оценка характеристик вариантов исполнения проекта

Объект исследования Критерии	Весовой коэффициент параметра	Исп. 1	Исп. 2
Функциональная мощность	0,15	5	4
Скорость работы	0,2	5	4
Надежность	0,3	5	5
Удобство эксплуатации	0,15	5	5
Точность распознавания	0,2	5	4
Итого	1	5	4,45

$$I_{p-исп1} = 5 \cdot 0,15 + 5 \cdot 0,2 + 5 \cdot 0,3 + 5 \cdot 0,15 + 5 \cdot 0,2 = 5$$

$$I_{p-исп2} = 4 \cdot 0,15 + 4 \cdot 0,2 + 5 \cdot 0,3 + 5 \cdot 0,15 + 4 \cdot 0,2 = 4,45$$

Интегральный показатель эффективности вариантов исполнения разработки определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{\text{исп.1}} = \frac{I_{p-\text{исп1}}}{I_{\text{финр}}^{\text{исп 1}}}$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных. Сравнительная эффективность проекта $\mathcal{E}_{\text{ср}}$ находится по формуле:

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{исп.1}}}{I_{\text{исп.2}}}$$

Результаты полученных показателей занесем в Таблицу 15.

Табл. 15 Расчет сравнительной эффективности вариантов исполнения

Исполнение	$I_{p-\text{исп}}$	$I_{\text{финр}}^{\text{исп.}}$	$I_{\text{исп.}}$	$\mathcal{E}_{\text{ср}}$	
				Исп. 1	Исп. 2
Исполнение 1	5	1	5	1	1,13
Исполнение 2	4,45	0,79	5,63	0,89	1

Проанализировав полученные данные, можно говорить о том, что наиболее эффективным вариантом исполнения проекта будет исполнение 2, которое на 13% эффективнее варианта исполнения 1.

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Институт	Институт Кибернетики
Направление подготовки (специальность)	Мехатроника и робототехника
Уровень образования	Бакалавриат
Кафедра	Систем управления и мехатроники
Период выполнения	весенний семестр 2016/2017 учебного года

Студенту:

Группа	ФИО
8Е31	Кремлев Иван Андреевич

Тема работы:

Создание программной компоненты по распознаванию дорожных знаков в сложных погодных условиях на основе каскадного классификатора	
Утверждена приказом проректора-директора (директора) (дата, номер)	№3935/с от 05.06.2017

Форма представления работы:

Бакалаврская работа
(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

ЗАДАНИЕ

Исходные данные к разделу «Социальная ответственность»:	1. Знакомство и отбор законодательных и нормативных документов по теме.
Перечень вопросов, подлежащих исследованию, проектированию и разработке	1. Экологическая безопасность - Анализ воздействия объекта на атмосферу (выбросы); - Анализ возможной утилизации объектов; - Разработка возможного решения проблемы. 2. Электромагнитное излучение - Физико – химическая природа фактора, его связь с разрабатываемой темой; - Воздействие фактора на организм человека; - Приведение допустимых норм с необходимой размерностью; - Предлагаемые средства защиты. 3. Некорректная работа программы - Анализ возможных чрезвычайных ситуаций, которые могут возникнуть при эксплуатации системы; - Возможные решения для чрезвычайных ситуаций. 4. Искусственный интеллект - Анализ влияние искусственного интеллекта на общество. 5. Правовые и организационные вопросы обеспечения безопасности - Специальные правовые нормы трудового законодательства.;
Дата выдачи задания для раздела по линейному графику	18.03.2017

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. ЭБЖ	Невский Е.С.	-		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8Е31	Кремлев Иван Андреевич		

5. Социальная ответственность

Введение

В данном разделе ВКР рассматриваются вопросы, которые касаются и правил в процессе использования разработанной программной компоненты и соблюдения санитарных норм. Были рассмотрены меры по защите работника от возможного негативного воздействия среды, а также были исследованы опасные и вредные факторы среды. Кроме того, были рассмотрены вопросы охраны окружающей среды от негативного воздействия программной компоненты и возможные чрезвычайные ситуации и действия, которые работник должен выполнить в случае возникновения чрезвычайной ситуации.

Разработанная система распознавания дорожных знаков будет использоваться в автомобиле. Работа сотрудника при использовании программного обеспечения может быть классифицирована, как работа высокой точности. Были рассмотрены и выделены такие вредные факторы как: неправильное освещение, высокий уровень электромагнитного излучения. К опасным факторам при работе с персональным компьютером можно отнести высокое напряжение и возможность короткого замыкания, влекущего за собой опасность поражения рабочего электрическим током. Рассматриваются вопросы правового регулирования трудовых отношений, связанных с использованием разработанной системы.

Программные продукты не оказывают непосредственного негативного влияния на окружающую среду, в отличие от реально существующих объектов, таких, как, например, атомная электростанция, однако ошибки в проектировании или реализации программных продуктов способны оказать значительное влияние на безопасность объектов, которые управляются с помощью ПО, а также на безопасность пользователей ПО (частных лиц или организаций).

5.1 Экологическая безопасность

Одними из наиболее сильно загрязняющих веществ при эксплуатации беспилотного автомобиля являются выхлопные газы и нефтепродукты при их испарении. Из 35 млн. тонн вредных выбросов 89% приходится на выбросы автомобильного транспорта и предприятий дорожно-строительного комплекса, так же транспорт является одним из основных источников шума в городах и вносит значительный вклад в тепловое загрязнение окружающей среды. Выбросы от автомобильного транспорта в России составляют около 22 млн. тонн в год. Для того, чтобы снизить влияние данного фактора на экологию, необходимо так сконструировать беспилотный автомобиль, чтобы он меньше загрязнял атмосферный воздух токсичными компонентами отработавших газов и создавал бы шум более низкого уровня. Также важным является использование средств и методов организации и регулирования движения, обеспечивающих оптимальные режимы движения и характеристики транспортных потоков, сокращение остановок у светофоров, числа переключения передач и времени работы двигателей на неустановившихся режимах. Это позволит как снизить расход топлива, так и снизить выброс загрязняющих веществ в атмосферу. [29]

Одним из наиболее оптимальных вариантов по экономии топлива является система «робот-человек». С помощью так называемых актуаторов, которые, получая сигналы от компьютера, переключают передачи вместо водителя, можно снизить выброс выхлопных газов. Когда именно переключать и на какую передачу — решает электроника в зависимости от оборотов двигателя и нагрузки. Главная задача состоит в обеспечении плавного изменения крутящего момента от двигателя, что делает разгон автомобиля плавным, без рывков и провалов.

Для работы с системой распознавания дорожных знаков сотруднику необходим компьютер, следовательно, в нашем случае воздействие на окружающую среду исходит от персонального компьютера. Ноутбук может нагреваться, а также быть источником электромагнитного и ионизирующего излучения, а

также шума. Для защиты здоровья сотрудников, работающих с программным обеспечением, рекомендуется соблюдать необходимую дистанцию при работе с компьютером (от 0.5 до 1 м), а также использовать эффективные охлаждающие системы и наиболее современное и эффективное оборудование. [30]

При необходимости утилизировать вышедшую из употребления электронику наиболее безопасным для окружающей среды способом необходимо обращаться в специализированную компанию по утилизации. Такие компании действуют на всей территории Российской Федерации, в том числе и в Томской области. Необходимо отметить, что в целом при работе с компьютером существенного загрязнения окружающей среды не происходит и вредные выбросы не сравнимы с производственными.

5.2 Электромагнитное излучение

Электромагнитным излучением называется распространяющееся в пространстве возмущение (изменение состояния) электромагнитного поля. Источником излучения могут стать ноутбук или веб-камера. Негативное воздействие электромагнитного излучения на организм человека проявляется в повышении температуры тела, а также может спровоцировать микропроцессы в организме, изменяющие свойства молекул. Длительное воздействие повышенного уровня электромагнитного излучения повышает утомляемость, может вызвать гипертонию, катаракту, изменения в крови, повышает сонливость. В ГОСТ Р 12.1.019-2009 регулируются максимально допустимые уровни напряженности электрического и магнитного полей, плотности потока энергии электромагнитного поля. Соответствующие значения показаны в таблице 16.

Табл. 16 ПДУ энергетических экспозиций ЭМП диапазона частот ≥ 30 кГц - 300 ГГц

Диапазоны частоты	Предельно допустимые уровни энергетической экспозиции		
	По электрической составляющей, $(В/м)^2 \cdot ч$	По магнитной составляющей, $(А/м)^2 \cdot ч$	По плотности потока энергии, $(мкВт/см^2) \cdot ч$
30 кГц - 3 МГц	20000	200	-
3 - 30 МГц	7000	-	-
30 - 50 МГц	800	0,72	-
50 - 300 МГц	800	-	-
300 МГц - 300 ГГц	-	-	200

5.3 Работа в условиях чрезвычайных ситуаций

Под ошибками в работы программного обеспечения понимается либо отличное от задуманного функционирование различных компонентов системы, либо же полное отсутствие требуемого функционала как следствие ошибок в реализации приложения. Подобные проблемы могут проявляться из-за ошибок в коде приложения, которая взаимодействует с базой данных, или же ошибочная структура базы данных обучающей выборки сама по себе.

Допустим, наша система дала сбой, при этом программная компонента функционирует на бензовозе. Распознать дорожный знак «Движение без остановки запрещено» не удалось и грузовик продолжил свое движение. Потеряв управление и не имея возможности перейти в ручной режим, бензовоз врывается в автозаправочную станцию. В результате взрыва могут пострадать работники автозаправочной станции, а также ее клиенты. Для того, чтобы подобные ситуации не возникали, необходимо, предусмотреть постоянное нахождение води-

теля в салоне бензовоза. В случае возникновения чрезвычайной ситуации, человек может взять управление на себя, тем самым можно предотвратить подобные ситуации.

Еще одним примером нештатной ситуации может служить выезд беспилотного автомобиля на встречную полосу, в связи с объездом препятствия. Последствия подобной ситуации могут зависеть от многих факторов. Они могут как ограничиться небольшим сбоем в программе, так и привести к трагическим последствиям. Среди основных можно выделить следующие:

1. Дорожные знаки.
2. Интенсивность движения.
3. Дорожное покрытие.
4. Погодные условия.
5. Видимость дороги.
6. Характер перевозимого груза или пассажиров.
7. Наличие пешеходов.
8. Наличие ДТП.
9. Техническое состояние автомобиля.

На данный момент беспилотные транспортные средства не в состоянии учитывать все многообразие ситуаций, возникающих на дороге. Поэтому, необходимо продумать алгоритм движения беспилотного транспортного средства и систему распознавания дорожных знаков, которая будет работать без сбоев.

5.4 Искусственный интеллект

Научно-технический прогресс – неотъемлемая и важная часть современной жизни. Техника занимает все больше времени и места, развивается подобно живым организмам, за исключением того, что все эти процессы в технике контролируются человеком. Однако, есть и негативные стороны прогресса, которые обязывают общество задумываться над проблемами отчуждения человека. Рассмотрим влияние программной компоненты на общество, а точнее на развитие технологий разработки искусственного интеллекта в целом.

Одна из особенностей разрабатываемой программной компоненты - каскадный классификатор, который является одним из методов искусственного интеллекта. Принцип обучения интеллекта и способ его «мышления» во многом схож с человеком: обучается ИИ посредством использования обучающей выборки. Интеллект изначально является абсолютно «неразумным», но в процессе обучения он получает некий опыт, основываясь на котором в последующем он будет принимать решения. Однако, в рамках данной работы получить полноценный интеллект, который смог бы хотя бы отдалённо напоминать человеческое сознание довольно проблематично, ведь деятельность ИИ ограничится только принятием решений при распознавании дорожных знаков. Поэтому непосредственно сам ИИ, полученный в данной работе, не будет каким-либо образом влиять на общество. Он не сможет внезапно выйти из-под контроля и начать самостоятельно обучаться сторонним задачам, поэтому непосредственное негативное влияние интеллекта на общество практически равно нулю. А те недовольства, которые могли бы возникнуть в обществе, можно легко решить, сделав технологию ИИ как можно более открытой, чтобы общество могло удостовериться в полной безопасности данной работы.

Но рассматриваемая работа имеет и другой эффект: для того, чтобы разработать ИИ, были изучены и протестированы различные методы разработки и обучения искусственных интеллектов. Все эти методы могут быть применимы в любой другой области человеческой деятельности. Более того, данные разработки могут применять при создании искусственного сознания, подобного человеческому разуму. Именно этот аспект работы и имеет существенное влияние на общество. Например, разработка полноценного ИИ позволит создавать роботов, которые смогут заменить людей во многих областях человеческой деятельности и множество людей потеряют свои рабочие места. Однако, всё это можно легко решить открытием новых видов деятельности, в которых роботы не смогут заменить человека: например, обслуживание роботов, либо создание

новых их версий. Но до подобного развития событий человечеству ещё достаточно далеко, хотя работы, такие как рассматриваемая здесь, могут значительно приблизить человечество к созданию совершенного искусственного интеллекта. Польза от разработки и совершенствования ИИ ощутима уже сейчас, работы в данном направлении крайне перспективны и прибыльны. А проблема ИИ решается достаточно легко: необходимо переключиться на разработку не «универсального», а узкоспециализированного искусственного разума, который сможет решать лишь крайне ограниченный круг задач. Примером может послужить искусственный интеллект в данной работе, который решает только одну, узкоспециализированную задачу – распознавание дорожных знаков.

5.5 Правовые и организационные вопросы обеспечения безопасности

Эксплуатация программной компоненты регулируется следующими нормативными актами:

1. СанПиН 2.2.4.1191-03. Электромагнитные поля в производственных условиях.
2. ГОСТ Р 12.1.019-2009. Система стандартов безопасности труда (ССБТ).
3. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
4. ГОСТ Р 53692-2009. Ресурсосбережение. Обращение с отходами.

Заключение

Основной целью работы была программной компоненты для распознавания дорожных знаков.

Для достижения этой цели перед исполнителем был поставлен ряд задач, касающихся изучения существующих методов выделения объектов на изображении, поиска области знака на изображениях, распознавания символов на знаках. Существующие методы поиска объектов на изображениях были изучены, и был реализован соответствующий алгоритм, позволяющий решить эту задачу.

Были изучены методы распознавания дорожных знаков, что также было реализовано в работе в виде приложения с использованием библиотеки OpenCV. Итоговая программная система способна распознавать основные из наиболее часто встречающихся видов дорожных знаков ограничения скорости.

Естественно, в реализации остаются места, которые возможно оптимизировать для лучшей работы, а также возможно реализовать распознавание нетривиальных (комбинированных) знаков, вид которых не позволяет распознавать их на данном этапе.

В программе реализованы все шаги, необходимые для корректной работы в различных условиях: возможность настраивать фильтры, автоматическое форматирование различных видов входных и выходных данных в соответствии с требованиями программы.

Список использованной литературы

1. Ваганов, В. И. Вождение автотранспортных средств / В.И. Ваганов, А.А. Рывкин. - М.: Транспорт, 2013. - 224 с.
2. Горбачев, М. Г. Безопасное вождение современного автомобиля / М.Г. Горбачев. - М.: Рипол Классик, 2017. - 256 с
3. Куперман, А. И. Безопасное управление автомобилем / А.И. Куперман. - М.: Транспорт, 2017. - 160 с.
4. Фейгин, З. И. Справочник автолюбителя / З.И. Фейгин, Р.П. Гнатюк-Данильчук. - М.: Ураджай, 2017. - 256 с.
5. Прэтт У. Цифровая обработка изображений. М.: Мир, 1982. - 311 с.
6. Форсайт Д.А., Понс Ж. Компьютерное зрение. Современный подход. – Вильямс, 2004. – 928 с.
7. A. Broggi, P. Cerri, P. Medici, P. Porta, and G. Ghisio, “Real time road signs recognition,” Intelligent Vehicles Symposium, 2007 IEEE, pp. 981–986, June 2007.
8. A. D. L. Escalera, J. M. A. Armingol, and M. Mata, “Traffic sign recognition and analysis for intelligent vehicles,” Image and Vision Computing, vol. 21, pp. 247–258, 2003.
9. X. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, “Recognition of traffic signs based on their colour and shape features extracted using human vision models,” Journal of Visual Communication and Image Representation, vol. 17, no. 4, pp. 675–685, 2006.
10. Ankit Desay. AdaBoost Extensions for Cost-Sensitive Classification. LAP Lambert Academic Publishing, 2012. 104 pp.
11. Якимов, П.Ю. Предварительная обработка цифровых изображений в системах локализации и распознавания дорожных знаков // Компьютерная оптика. –2013. – Том 37. - № 3. – С. 401-405.
12. Houben, S. Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark / S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, C. Igel // International Joint Conference on Neural Networks. – 2013.

13. Lopez, L. Color-based road sign detection and tracking. Image Analysis and Recognition / L. Lopez, O. Fuentes // Lecture Notes in Computer Science. – Springer. – 2007. – P. 1138-1147.
14. P. Viola and M. Jones, “Robust real-time object detection,” in International Journal of Computer Vision, 2001.
15. Фурсов, В.А. Локализация контуров объектов на изображениях при вариациях масштаба с использованием преобразования Хафа / С.А. Бибииков, В.А. Фурсов, П.Ю. Якимов // Компьютерная оптика. – 2013. – Т. 37, № 4. – С. 502-508.
16. Якимов, П. Исследование эффективности технологии CUDA в задаче распределённой предпечатной подготовки цифровых изображений / С.А. Бибииков, А.В. Никоноров, В.А. Фурсов, П.Ю. Якимов // Сборник трудов «Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность». – 2009. – С. 21-26.
17. Belaroussi, R. Road Sign Detection in Images / R. Belaroussi, P. Foucher, J.P. Tarel, B. Soheilian, P. Charbonnier, N. Paparoditis // A Case Study, 20th International Conference on Pattern Recognition (ICPR) – 2010. – P. 484-488.
18. Л. Шапиро, Дж. Стокман. Компьютерное зрение, 2006. 752 с.
19. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Издание 3-е, исправленное и дополненное Москва: Техносфера, 2012. – 1104 с.
20. Koschan, A. Digital Color Image Processing / A. Koschan, M.A. Abidi // ISBN 978-0-470-14708-5. – 2008. – 376 P.
21. Travis, D. Effective Color Displays Theory and Practice // Academic Press, ISBN 0-12-697690-2. – 1991. – 328 p.
22. Бьерн Страуструп. Язык программирования C++. М.: Бином, 2015. 1136 с.
23. Гэри Брадски. Изучение OpenCV. O'Reilly Media, 2008. 556 с.
24. С. Robin, “Train your own OpenCV HAAR classifier”. [Режим доступа]: <http://codingrobin.de/2013/07/22/trainyour-own-opencv-haarclassifier.html>.

25. N. Seo, “OpenCV haartraining (Rapid Object Detection With A Cascade of Boosted Classifiers Based on Haar-like Features),”. [Режим доступа]: <http://note.sonots.com/SciSoftware/haartraining.html>.
26. Ценина Т.Т. Организация и техника внешнеторговых операций: Учебное пособие.— СПб.: Изд-во СПбГУЭФ, 2009. – 200 с.
27. Г.Я. Гольдштейн Стратегические аспекты управления НИОКР. Монография. Таганрог: Изд-во ТРТУ, 2010. 244с.
28. Виленский П.П. Оценка эффективности инвестиционных проектов. Академия народного хозяйства, 2008. -1104 с.
29. В.Ю. Микрюков. Безопасность жизнедеятельности: учебник. — Москва : КноРус, 2016. — 283 с.
30. СанПиН 2.2.4.1191-03. Электромагнитные поля в производственных условиях.

Приложение А

Каскадный классификатор

```
<?xml version="1.0"?>
<opencv_storage>
<cascade>
  <stageType>BOOST</stageType>
  <featureType>HAAR</featureType>
  <height>20</height>
  <width>20</width>
  <stageParams>
    <boostType>GAB</boostType>
    <minHitRate>9.9500000476837158e-001</minHitRate>
    <maxFalseAlarm>4.0000000596046448e-001</maxFalseAlarm>
    <weightTrimRate>9.4999999999999996e-001</weightTrimRate>
    <maxDepth>1</maxDepth>
    <maxWeakCount>100</maxWeakCount></stageParams>
  <featureParams>
    <maxCatCount>0</maxCatCount>
    <featSize>1</featSize>
    <mode>ALL</mode></featureParams>
  <stageNum>15</stageNum>
  <stages>
    <!-- stage 0 -->
    <_>
      <maxWeakCount>4</maxWeakCount>
      <stageThreshold>-1.6327016353607178e+000</stageThreshold>
      <weakClassifiers>
        <_>
          <internalNodes>
            0 -1 33 -1.8862386047840118e-001</internalNodes>
          <leafValues>
            5.4696130752563477e-001 -7.6878613233566284e-001</leafValues></_>
        <_>
          <internalNodes>
```



```

    0 -1 4 1.1436487548053265e-002</internalNodes>
    <leafValues>
    -6.4654630422592163e-001 5.9246361255645752e-001</leafValues></_>
  <_>
    <internalNodes>
    0 -1 112 1.4806104823946953e-002</internalNodes>
    <leafValues>
    -7.7234685420989990e-001 4.7075948119163513e-001</leafValues></_>
  <_>
    <internalNodes>
    0 -1 96 -2.0883162505924702e-004</internalNodes>
    <leafValues>
    5.1788955926895142e-001 -6.8812870979309082e-001</leafValues></_></weakClassi-
fiers></_>
  <!-- stage 1 -->
  <_>
    <maxWeakCount>5</maxWeakCount>
    <stageThreshold>-1.3822076320648193e+000</stageThreshold>
    <weakClassifiers>
    <_>
      <internalNodes>
      0 -1 52 4.6981547027826309e-002</internalNodes>
      <leafValues>
      -8.2743364572525024e-001 2.9838711023330688e-001</leafValues></_>
    <_>
      <internalNodes>
      0 -1 27 4.3758009560406208e-003</internalNodes>
      <leafValues>
      -6.3025408983230591e-001 5.8225786685943604e-001</leafValues></_>
    <_>
      <internalNodes>
      0 -1 18 -1.8302239477634430e-002</internalNodes>
      <leafValues>
      3.8388404250144958e-001 -8.6029422283172607e-001</leafValues></_>

```

```

<_>
  <internalNodes>
    0 -1 11 1.8881820142269135e-003</internalNodes>
  <leafValues>
    -7.3129785060882568e-001 3.6285561323165894e-001</leafValues></_>
<_>
  <internalNodes>
    0 -1 105 -8.7470863945782185e-005</internalNodes>
  <leafValues>
    4.4181787967681885e-001 -6.7125958204269409e-001</leafValues></_></weakClassi-
fiers></_>
<!-- stage 2 -->
<_>
  <maxWeakCount>5</maxWeakCount>
  <stageThreshold>-1.1364247798919678e+000</stageThreshold>
  <weakClassifiers>
    <_>
      <internalNodes>
        0 -1 55 3.0422400683164597e-002</internalNodes>
      <leafValues>
        -9.0654206275939941e-001 3.2352942228317261e-001</leafValues></_>
    <_>
      <internalNodes>
        0 -1 77 -1.1580903083086014e-002</internalNodes>
      <leafValues>
        5.3339087963104248e-001 -5.7019501924514771e-001</leafValues></_>
    <_>
      <internalNodes>
        0 -1 115 -7.3941331356763840e-004</internalNodes>
      <leafValues>
        3.8492733240127563e-001 -7.8562790155410767e-001</leafValues></_>
    <_>
      <internalNodes>
        0 -1 26 9.8137129098176956e-003</internalNodes>

```

```

    <leafValues>
      -3.5704827308654785e-001 8.4422636032104492e-001 </leafValues> </_>
    <_>
    <internalNodes>
      0 -1 49 2.2211161255836487e-001 </internalNodes>
    <leafValues>
      3.1243321299552917e-001 -1. </leafValues> </_> </weakClassifiers> </_>
<!-- stage 3 -->
<_>
<maxWeakCount>8</maxWeakCount>
<stageThreshold>-1.4127116203308105e+000</stageThreshold>
<weakClassifiers>
  <_>
    <internalNodes>
      0 -1 3 -3.3135283738374710e-002 </internalNodes>
    <leafValues>
      7.3469388484954834e-001 -6.1794018745422363e-001 </leafValues> </_>
  <_>
    <internalNodes>
      0 -1 81 -2.7839591726660728e-002 </internalNodes>
    <leafValues>
      6.9340646266937256e-001 -4.0550678968429565e-001 </leafValues> </_>
  <_>
    <internalNodes>
      0 -1 75 3.1563533411826938e-005 </internalNodes>
    <leafValues>
      -7.2958868741989136e-001 3.7139376997947693e-001 </leafValues> </_>
  <_>
    <internalNodes>
      0 -1 51 3.1460404396057129e-002 </internalNodes>
    <leafValues>
      -7.2197884321212769e-001 3.4046509861946106e-001 </leafValues> </_>
  <_>
    <internalNodes>

```

```

    0 -1 108 1.3264226727187634e-003</internalNodes>
    <leafValues>
    3.5295933485031128e-001 -6.6770291328430176e-001</leafValues></_>
  <_>
    <internalNodes>
    0 -1 67 -5.8512100949883461e-003</internalNodes>
    <leafValues>
    6.9002526998519897e-001 -4.0525573492050171e-001</leafValues></_>
  <_>
    <internalNodes>
    0 -1 9 4.4461982324719429e-003</internalNodes>
    <leafValues>
    -2.9503583908081055e-001 6.6148608922958374e-001</leafValues></_>
  <_>
    <internalNodes>
    0 -1 110 1.1742194183170795e-003</internalNodes>
    <leafValues>
    3.0865260958671570e-001 -7.3873716592788696e-001</leafValues></_></weakClassi-
fiers></_>
  <!-- stage 4 -->
  <_>
    <maxWeakCount>8</maxWeakCount>
    <stageThreshold>-1.5045621395111084e+000</stageThreshold>
    <weakClassifiers>
    <_>
      <internalNodes>
      0 -1 2 -1.2427415698766708e-002</internalNodes>
      <leafValues>
      8.6301368474960327e-001 -5.7894736528396606e-001</leafValues></_>
    <_>
      <internalNodes>
      0 -1 84 4.9863746389746666e-003</internalNodes>
      <leafValues>
      -3.5204887390136719e-001 7.9394853115081787e-001</leafValues></_>

```

```

<_>
  <internalNodes>
    0 -1 88 4.8526255413889885e-003</internalNodes>
  <leafValues>
    -3.5775437951087952e-001 5.8134990930557251e-001</leafValues></_>
<_>
  <internalNodes>
    0 -1 63 2.0323660224676132e-002</internalNodes>
  <leafValues>
    -2.2215421497821808e-001 7.9634881019592285e-001</leafValues></_>
<_>
  <internalNodes>
    0 -1 11 4.7522112727165222e-003</internalNodes>
  <leafValues>
    -5.3699457645416260e-001 3.9329338073730469e-001</leafValues></_>
<_>
  <internalNodes>
    0 -1 58 3.9585400372743607e-003</internalNodes>
  <leafValues>
    -3.4242072701454163e-001 6.8539816141128540e-001</leafValues></_>
<_>
  <internalNodes>
    0 -1 118 -1.3845504145137966e-004</internalNodes>
  <leafValues>
    4.2903533577919006e-001 -5.0285607576370239e-001</leafValues></_>
<_>
  <internalNodes>
    0 -1 15 -6.5888285462278873e-005</internalNodes>
  <leafValues>
    4.5832607150077820e-001 -4.8632442951202393e-001</leafValues></_></weakClassi-
fiers></_>
<!-- stage 5 -->
<_>
  <maxWeakCount>8</maxWeakCount>

```

Приложение Б

Матрицы SWOT-анализа

Интерактивная матрица слабых сторон и возможностей проекта

Возможности проекта		Сл1	Сл2	Сл3
	B1	-	-	+
	B2	-	+	+
	B3	-	+	+
	B4	-	-	+

Интерактивная матрица сильных сторон и угроз проекта

Угрозы проекта		С1	С2	С3	С4
	У1	-	+	+	+
	У2	-	-	-	-
	У3	+	+	-	-
	У4	+	-	-	+

Интерактивная матрица слабых сторон и угроз проекта

Угрозы проекта		Сл1	Сл2	Сл3	Сл4
	У1	-	+	+	-
	У2	+	-	-	-
	У3	-	+	+	-
	У4	-	-	+	+